

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-159009

(43)Date of publication of application : 31.05.2002

(51)Int.Cl. H04N 7/30
H03M 7/30
H03M 7/40

(21)Application number : 2001-261489 (71)Applicant : CANON INC
(22)Date of filing : 30.08.2001 (72)Inventor : DOMINICK IPPU

(30)Priority

Priority number : 2000 PQ9824 Priority date : 01.09.2000 Priority country : AU

(54) METHOD AND DEVICE FOR ENTROPY CODING AND DECONDING

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an entropy coding method by which symbols indicating code blocks having the transformation coefficients of a digital image can be entropy coded.

SOLUTION: In this entropy coding method, a significance propagation path 314, a magnitude refinement path 316, and a cleanup path 318 are used for entropy coding the symbols. In the method, a first coefficient list of the positions in code blocks having a symbol that becomes the object of entropy coding in the significance propagation path 314 of the present bit plane, and a second list of the positions of the coefficients having a symbol that becomes the object of the entropy coding in the magnitude refinement path 316 of the present bit plane, are generated. In addition, a third list of the positions of the coefficients which have a symbol that becomes the object of entropy coding in the cleanup path 318 of the present bit plane and are the factors in the code blocks is also generated.

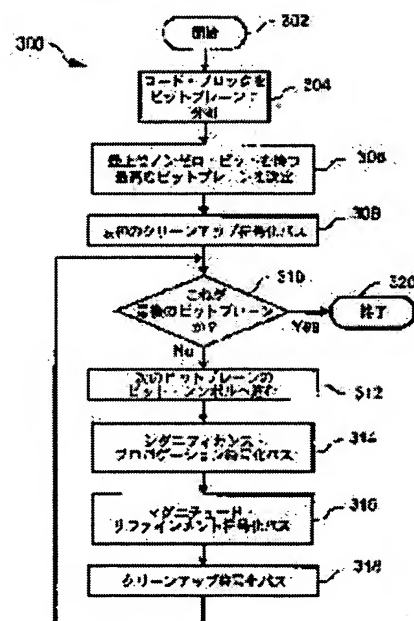


Fig. 3

LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-159009

(P2002-159009A)

(43) 公開日 平成14年5月31日 (2002.5.31)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
H 0 4 N 7/30		H 0 3 M 7/30	A 5 C 0 5 9
H 0 3 M 7/30		7/40	5 J 0 6 4
7/40		H 0 4 N 7/133	Z

審査請求 未請求 請求項の数38 O L (全 40 頁)

(21) 出願番号 特願2001-261489(P2001-261489)

(22) 出願日 平成13年8月30日 (2001.8.30)

(31) 優先権主張番号 P Q 9 8 2 4

(32) 優先日 平成12年9月1日 (2000.9.1)

(33) 優先権主張国 オーストラリア (AU)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 ドミニック イップ

オーストラリア国 2113 ニュー サウス

ウェールズ州, ノース ライド, ト

ーマス ホルト ドライブ 1 キヤノン

インフォメーション システムズ リサ

ーチ オーストラリア プロプライエタリ

ー リミテツド 内

(74) 代理人 100076428

弁理士 大塚 康徳 (外3名)

最終頁に続く

(54) 【発明の名称】 エントロピ符号化及び復号化方法とその装置

(57) 【要約】 (修正有)

【課題】 デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化する方法が提供される。

【解決手段】 シンボルをエントロピ符号化するための、シグニフィカンス・プロパゲーション・パス314と、マグニチュード・リファインメント・パス316と、クリーンアップ・パス318とを有する。現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中エントロピ符号化の対象となるシンボルを持つコード・ブロック内の位置の第1の係数リストを生成し、コード・ブロック内の前記係数である、現在のビットプレーンのマグニチュード・リファインメント・パス中エントロピ符号化の対象となるシンボルを持つ係数の位置の第2のリストも生成し、コード・ブロック内の前記係数である、現在のビットプレーンのクリーンアップ・パス中にエントロピ符号化の対象となるシンボルを持つ係数の位置の第3のリストをさらに生成する。

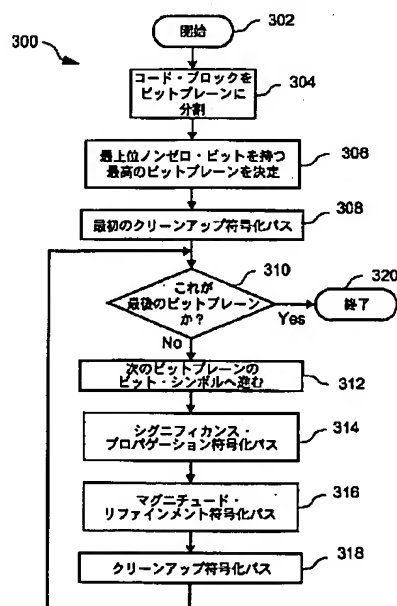


Fig. 3

【特許請求の範囲】

【請求項1】 デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化する方法であって、第1のビットプレーンから所定の最下位ビットプレーンまでの各ビットプレーンについて、

現在のビットプレーン内にゼロの有意状態を持つ係数であって、現在のビットプレーン内に1の有意状態を持つ近傍係数を持つ係数に属するシンボルである、現在のビットプレーン内の全てのシンボルのエントロピ符号化を行うシグニフィカンス・プロパゲーション(significance propagation)ステップと、

前回のビットプレーン内に1の有意状態を持つ係数に属するシンボルである、現在のビットプレーン内の全てのシンボルのエントロピ符号化を行うマグニチュード・リファインメント・ステップと、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップ又は前記マグニチュード・リファインメント・ステップ中、それ以前にエントロピ符号化されなかったシンボルである、現在のビットプレーン内の全てのシンボルのエントロピ符号化を行うクリーンアップ・ステップとを実行し、

更に、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップ中、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第1のリストを、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップに先行して生成するステップと、

現在のビットプレーンの前記マグニチュード・リファインメント・ステップ中、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第2のリストを、現在のビットプレーンの前記マグニチュード・リファインメント・ステップに先行して生成するステップと、

現在のビットプレーンの前記クリーンアップ・ステップ中、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第3のリストを、現在のビットプレーンの前記クリーンアップ・ステップに先行して生成するステップと、を有することを特徴とする方法。

【請求項2】 請求項1に記載の方法において、前記第1及び第2のリストを生成する前記ステップが、前回のビットプレーンの前記クリーンアップ・ステップ中に行われることを特徴とする方法。

【請求項3】 請求項1に記載の方法において、前記第3のリストを生成する前記ステップが、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップ中に行われることを特徴とする方法。

【請求項4】 請求項1に記載の方法において、現在のビットプレーンの前記シグニフィカンス・プロパ

ゲーション・ステップは、

前回のビットプレーンの前記クリーンアップ・ステップによって生成される前記第1のリスト内の前記位置に対応するシンボルである、現在のビットプレーン内のシンボルのエントロピ符号化を行うサブステップと、

ゼロの非有意状態を持つ、現在のビットプレーン内の任意のシンボルであって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップ中に1の有意状態に変化する近傍として、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップ中、予めエントロピ符号化されたシンボルを持つ係数を有する任意のシンボルである、所定の走査順で前記近傍に後続する任意のシンボルのエントロピ符号化を行うサブステップと、

ゼロの非有意状態を持つ、コードブロックの係数の位置と、1の有意状態を持つ現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップでエントロピ符号化されるシンボルを有する前記係数に隣接する係数の位置の第4のリストを生成するサブステップと、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップ中、1の有意状態に変化する前記係数の前記第1のリストの前記位置にタグを付けるサブステップと、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップ中、1の有意状態に変化する前記係数の前記第4のリストの前記位置にタグを付けるサブステップと、

前記コード・ブロック内の係数の位置の前記第3のリストを生成するサブステップと、を有することを特徴とする方法。

【請求項5】 請求項4に記載の方法において、

前記クリーンアップ・ステップは、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップによって生成されたシンボルであって、前記第3のリスト内の前記位置に対応するシンボルである、現在のビットプレーン内のシンボルのエントロピ符号化を行うサブステップと、

現在のビットプレーンの前記クリーンアップ・ステップ中、エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンの前記クリーンアップ中、1の有意状態に変化する係数である、前記コード・ブロック内の前記係数の位置の第5のリストを生成するサブステップと、

現在のビットプレーンの前記クリーンアップ・ステップ中、エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンの前記クリーンアップで、ゼロの有意状態を持つ係数である、前記コード・ブロック内の前記係数の位置の第6のリストを生成するサブステップと、を有することを特徴とする方法。

【請求項6】 請求項5に記載の方法において、

前記クリーンアップ・ステップは、
前記第1のリストを前記タグを付けられた位置の第7のリストと、タグを付けられなかった位置の第8のリストとにソートするサブステップと、
前記第4のリストを、前記タグを付けられた位置の第9のリストと、タグを付けられなかった位置の第10のリストとにソートするサブステップと、を有することを特徴とする方法。

【請求項7】 請求項6に記載の方法において、
現在のビットプレーンが使用される前記クリーンアップ・ステップは、
前記第2、第5、第7、第9のリストの前記位置を比較し併合して、現在のビットプレーンの後の次のビットプレーン用の位置の前記第2のリストを生成するサブステップと、
前記第6、第8、第10のリストの前記位置を比較し併合して、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストを生成するサブステップと、
前記生成された第1のリストから、1の有意状態を持つ係数である、前記コード・ブロック内の係数の任意の位置を除去するサブステップと、を有することを特徴とする方法。

【請求項8】 請求項1に記載の方法において、前記第1のビットプレーンは、有意係数を持つ前記コード・ブロックの最上位ビットプレーンより下の次のビットプレーンであることを特徴とする方法。

【請求項9】 請求項8に記載の方法において、
前記コード・ブロックの前記最上位ビットプレーン内の全てのシンボルをエントロピ符号化するの最初のクリーンアップ・ステップを有することを特徴とする方法。

【請求項10】 請求項9に記載の方法において、
前記最初のクリーンアップ・ステップは、
前記第1のビットプレーンに対する前記シグニフィカンス・プロパゲーション・ステップで、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の係数の位置の前記第1のリストを生成するサブステップと、
前記第1のビットプレーンに対する前記マグニチュード・リファインメント・ステップで、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の係数の位置の前記第2のリストを生成するサブステップと、を有することを特徴とする方法。

【請求項11】 請求項9に記載の方法において、
前記最初のクリーンアップ・ステップは、
前記最初のクリーンアップ・ステップ符号化パス中、符号化された前記各シンボルについて、前記エントロピ符号化されたシンボルが属する前記係数の前記有意状態を決定するサブステップであって、前記有意状態が1の場合、

第1のフラグをセットするサブステップと、
前記所定の1の有意状態を有する前記係数の前記近傍係数を決定し、前記各近傍係数について第2のフラグを1にセットするサブステップとを実行するサブステップと、

一方、前記所定の有意状態がゼロの場合、
前記第1のフラグをゼロにセットするサブステップと、
前記第1のビットプレーンが使用される前記マグニチュード・リファインメント・ステップ中、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の係数の位置の前記第2のリストを生成するサブステップであって、前記位置に関連する前記第1のフラグが1の場合、前記係数の前記位置が前記第2のリストに追加されるように構成されるサブステップと、
前記第1のビットプレーンが使用される前記シグニフィカンス・プロパゲーション・ステップ中、エントロピ符号化の対象となる前記シンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の前記第1のリストを生成するサブステップであって、前記位置に関連する前記第2のフラグが1であり、かつ、前記位置に関連する前記第1のフラグがゼロである場合、前記係数の前記位置を前記第1のリストに追加するように構成されるサブステップと、を有することを特徴とする方法。

【請求項12】 請求項1に記載の方法において、前記エントロピ符号化法が、前記シンボルのエントロピ符号化または復号化を行うことが可能であることを特徴とする方法。

【請求項13】 請求項1に記載の方法において、前記エントロピ符号化法が前記シンボルのエントロピ符号化を行う方法であって、現在のビットプレーン用の前記第1と、第2と、第3のリストを生成する前記ステップが、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップに先行して生成されることを特徴とする方法。

【請求項14】 請求項13に記載の方法において、現在のビットプレーンの前記第2のリストを生成する前記方法は、
現在のビットプレーン番号N以上の有意状態を持つ前記第2のリストに前記コード・ブロック内の係数の前記位置を追加するサブステップを有することを特徴とする方法。

【請求項15】 請求項13または14に記載の方法であって、現在のビットプレーンの前記第1のリストを生成する前記方法において、
現在のビットプレーン番号N未満の有意状態を持つ係数であって、所定の走査順でN以上の有意状態を持つ近傍係数係数に先行する係数である、前記コード・ブロック内の任意の係数の前記位置を前記第1のリストに追加するサブステップを有することを特徴とする方法。

【請求項16】 デジタル画像の変換係数を有するコー

ド・ブロックを表すシンボルをエントロピ符号化する装置において、

第1のビットプレーンから所定の最下位ビットプレーンへ、シンボルから成るビットプレーンに前記コード・ブロックを分割するビットプレーン・スプリックと、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中に、前記シンボルを符号化するエントロピ符号器と、

現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に、エントロピ符号化の対象となるシンボルを持つ係数である、コード・ブロック内の前記係数の位置の第1のリストを、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パスに先行して生成し、現在のビットプレーンのマグニチュード・リファインメント・パス中に、エントロピ符号化の対象となるシンボルを持つ係数である、コード・ブロック内の前記係数の位置の第2のリストを、現在のビットプレーンの前記マグニチュード・リファインメント・パスに先行して生成し、現在のビットプレーンのクリーンアップ・パス中に、エントロピ符号化の対象となるシンボルを持つ係数である、コード・ブロック内の前記係数の位置の第3のリストを、現在のビットプレーンの前記クリーンアップ・パスに先行して生成するリスト・メモリ・マネージャと、

前記第1の位置リストと、第2の位置リストと、第3の位置リストとを格納する記憶ユニットと、を有することを特徴とする装置。

【請求項17】 請求項16に記載の装置において、前記記憶ユニットは、前記リスト・メモリ・マネージャによってビットプレーン n の前記第1のリストを入力して格納し、さらに、既に生成され、格納されたビットプレーン $(n+1)$ の前記第1のリストを同時に出力するダブルバッファとなるように構成される第1の記憶ユニットと、前記リスト・メモリ・マネージャによってビットプレーン n の前記第2のリストを入力し、格納し、更に、既に生成され格納されたビットプレーン $(n+1)$ の前記第2のリストを同時に出力するダブルバッファとなるように構成される第2の記憶ユニットと、前記リスト・メモリ・マネージャによってビットプレーン n の前記第3のリストを入力し、格納し、更に、既に生成され格納されたビットプレーン $(n+1)$ の前記第3のリストを同時に出力するダブルバッファとなるように構成される第3の記憶ユニットと、を有することを特徴とする装置。

【請求項18】 請求項16に記載の装置において、前記リスト・メモリ・マネージャは、前回のビットプレーンのクリーンアップ・パス中に前記シンボルの前記エントロピ符号化を行っている間、前記第1及び第2のリス

トを生成することを特徴とする装置。

【請求項19】 請求項16に記載の装置において、前記リスト・メモリ・マネージャは、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に前記シンボルの前記エントロピ符号化を行っている間、前記第3のリストを生成することを特徴とする装置。

【請求項20】 請求項16に記載の装置において、前記エントロピ符号器は、前回のビットプレーンの前記クリーンアップ・パスによって生成された前記第1のリスト内の前記位置に対応する現在のビットプレーン内の前記シンボルのエントロピ符号化を、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に行う手段と、ゼロの非有意状態を持つ任意のシンボルであって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する近傍として予めエントロピ符号化されたシンボルを持つ近傍係数を有する任意のシンボルである、所定の走査順で前記近傍に後続する任意のシンボルのエントロピ符号化を、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に現在のビットプレーン内で行う手段と、を有し、

前記リスト・メモリ・マネージャは、ゼロの非有意状態を持つ係数であって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップ中、エントロピ符号化されるシンボルである、1の有意状態を持つシンボルを持つ係数に隣接する前記係数である、前記コード・ブロック内の前記係数の位置の第4のリストを生成する手段と、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に、1の有意状態に変化する前記係数の前記第1のリストの位置にタグを付ける手段と、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に、1の有意状態に変化する前記係数の前記第4のリストの位置にタグを付ける手段と、前記コード・ブロック内の係数の位置の前記第3のリストを生成する手段と、を有することを特徴とする装置。

【請求項21】 請求項20に記載の装置において、前記エントロピ符号器は、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パスによって生成されたシンボルである、前記第3のリスト内の前記位置に対応する現在のビットプレーン内のシンボルのエントロピ符号化を現在のビットプレーンのクリーンアップ・パス中に行う手段を有し、

前記リスト・メモリ・マネージャは、現在のビットプレーンの前記クリーンアップ・パス中、エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンの前記クリーンアップ中、1の有意状態に変化する係数である、前記コード・ブロック内

の前記係数の位置の第5のリストを生成する手段と、現在のビットプレーンの前記クリーンアップ・パス中にエントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンの前記クリーンアップ中ゼロの有意状態を持つ係数である、前記コード・ブロック内の前記係数の位置の第6のリストを生成する手段と、を有することを特徴とする装置。

【請求項22】 請求項21に記載の装置において、前記リスト・メモリ・マネージャは、タグを付けられた第7の位置リストと、タグを付けられなかった第8の位置リストとに前記第1のリストをソートする手段と、タグを付けられた第9の位置リストと、タグを付けられなかった第10の位置リストとに前記第4のリストをソートする手段と、を有することを特徴とする装置。

【請求項23】 請求項22に記載の装置において、前記リスト・メモリ・マネージャは、前記第2、第5、第7、第9のリストの前記位置を比較し併合して、現在のビットプレーンの後の次のビットプレーン用の位置の前記第2のリストを生成する手段と、前記第6、第8、第10のリストの前記位置を比較し併合して、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストを生成する手段と、前記生成された第1のリストから、1の有意状態を持つ係数である、前記コード・ブロック内の係数の任意の位置を取り除く手段と、を有することを特徴とする装置。

【請求項24】 請求項16に記載の装置において、前記第1のビットプレーンが、有意係数を持つ前記コード・ブロックの最上位ビットプレーンより下の次のビットプレーンであることを特徴とする装置。

【請求項25】 請求項24に記載の装置において、前記エントロピ符号器は、最初のクリーンアップ・ステップ・パス中に、前記コード・ブロックの前記最上位ビットプレーン内の全てのシンボルをエントロピ符号化することを特徴とする装置。

【請求項26】 請求項25に記載の装置において、前記リスト・マネージャは、前記第1のビットプレーン用として、前記シグニフィカンス・プロパゲーション・パス中、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の前記第1のリストを前記最初のクリーンアップ・ステップ・パス中に生成する手段と、前記第1のビットプレーン用の前記マグニチュード・リファインメント・パス中、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の前記第2のリストを、前記最初のクリーンアップ・ステップ・パス中生成する手段と、を有することを特徴とする装置。

【請求項27】 請求項25に記載の装置において、

前記リスト・マネージャは、前記最初のクリーンアップ・ステップ符号化パス中、符号化された前記各シンボルについて、前記エントロピ符号化されたシンボルが属する前記係数の前記有意状態を決定する手段を有し、前記決定手段は、前記有意状態が1であれば、第1のフラグを1にセットする手段と、前記有意状態が1であれば、前記所定の1の有意状態を有する前記係数の前記近傍係数を決定し、前記各近傍係数について第2のフラグを1にセットする手段と、前記所定の有意状態がゼロであれば、前記第1のフラグをゼロにセットする手段と、を有するように構成される決定手段と、前記第1のビットプレーン用の前記マグニチュード・リファインメント・パス中、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の前記第2のリストを前記最初のクリーンアップ・ステップ符号化パス中生成する手段を有し、前記位置に関連する前記第1のフラグが1の場合、前記係数の前記位置が前記第2のリストに追加されるように構成される手段と、

前記第1のビットプレーンについて前記シグニフィカンス・プロパゲーション・パス中、エントロピ符号化される前記シンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の前記第1のリストを生成する手段を有し、前記位置に関連する前記第2のフラグが1であり、かつ前記位置に関連する前記第1のフラグがゼロである場合、前記係数の前記位置を前記第1のリストに追加することを特徴とする装置。

【請求項28】 請求項16に記載の装置において、エントロピ符号化の前記装置が前記シンボルのエントロピ符号化または復号を行うことが可能であることを特徴とする装置。

【請求項29】 請求項16に記載の装置において、エントロピ符号化の前記装置が前記シンボルのエントロピ符号化装置であり、前記リスト・メモリ・マネージャが現在のビットプレーン用の前記第1、第2、第3のリストを現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パスに先行して生成することを特徴とする装置。

【請求項30】 請求項29に記載の装置において、前記リスト・メモリ・マネージャは、現在のビットプレーン番号N以上の有意状態を持つ前記第2のリストに前記コード・ブロック内の係数の前記位置を追加する手段を有することを特徴とする装置。

【請求項31】 請求項29に記載の装置において、前記リスト・メモリ・マネージャは、現在のビットプレーン番号N未満の有意状態を持ち、かつ、所定の走査順でN以上の有意状態を持つ係数である、前記コード・ブロック内の任意の係数の前記位置を近傍係数に先行する前

記第1のリストに追加する手段を有することを特徴とする装置。

【請求項32】 デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化するコンピュータ・プログラムを有するコンピュータ・プログラム製品であって、
前記コンピュータ・プログラムは、
第1のビットプレーンから所定の最下位ビットプレーンへシンボルから成るビットプレーンに前記コード・ブロックを分割するためのコードと、
シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中に前記シンボルをエントロピ符号化するためのコードと、
現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第1のリストを、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パスに先行して生成するためのコードと、
現在のビットプレーンのマグニチュード・リファインメント・パス中、エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第2のリストを、現在のビットプレーンの前記マグニチュード・リファインメント・パスに先行して生成するためのコードと、
現在のビットプレーンのクリーンアップ・パス中、エントロピ符号化の対象となるシンボルを有する、前記コード・ブロック内の前記係数の位置の第3のリストを、現在のビットプレーンの前記クリーンアップ・パスに先行して生成するためのコードと、
前記第1の位置リスト、第2の位置リスト、第3の位置リストとを格納するためのコードと、を有することを特徴とするコンピュータ・プログラム製品。

【請求項33】 係数を有するブロックを表すシンボルのエントロピ符号化方法であって、
第1のビットプレーンから所定の最小ビットプレーンまでの各ビットプレーンについて、
前記ブロックの現在のビットプレーン中、エントロピ符号化されるシンボルを有する、ブロック内の前記係数の位置の少なくとも1つのリストを生成するステップと、
前記少なくとも1つのリストを用いて、前記ブロックの現在のビットプレーン内の係数の前記シンボルをエントロピ符号化するステップと、を実行することを特徴とする方法。

【請求項34】 係数を有するブロックを表すシンボルをエントロピ符号化する装置であって、
前記ブロックの現在のビットプレーン中、エントロピ符号化されるシンボルを有する、前記ブロック内の前記係数の位置の少なくとも1つのリストを生成する生成ユニ

ットと、

前記少なくとも1つのリストを用いて、前記ブロックの現在のビットプレーン内の係数の前記シンボルをエントロピ符号化するエントロピ符号器と、を有することを特徴とする装置。

【請求項35】 係数を有するブロックを表すシンボルをエントロピ符号化するコンピュータ・プログラムを有するコンピュータ・プログラム製品であって、
前記ブロックの現在のビットプレーン中、エントロピ符号化されるシンボルを有する、前記ブロック内の前記係数の位置の少なくとも1つのリストを生成するためのコードと、
前記少なくとも1つのリストを用いて、前記ブロックの現在のビットプレーン内の係数の前記シンボルをエントロピ符号化するためのコードと、を有することを特徴とするコンピュータ・プログラム製品。

【請求項36】 デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化する方法であって、
第1のビットプレーンから所定の最下位ビットプレーンまでの各ビットプレーンについて、
現在のビットプレーン内でエントロピ符号化の対象となる前記シンボルをそれぞれ持つ係数の、第1、第2及び第3の位置リストに従って、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中に、前記コード・ブロックの現在のビットプレーンの前記変換係数のシンボルを符号化するステップと、
現在のビットプレーンの後の次のビットプレーン用の位置の第1のリストと、現在のビットプレーンの後の次のビットプレーン用の位置の第2のリストと、現在のビットプレーン用の位置の第3のリストとを生成するステップと、を有し、
前記生成するステップは、

ゼロの非有意状態を持つ係数であって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されるシンボルである、1の有意状態を持つシンボルを持つ係数に隣接する前記係数である、前記コード・ブロック内の前記係数の位置の第4のリストを、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に生成するサブステップと、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数である、現在のビットプレーン内の係数の位置の第1のリストの位置に、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パスでタグを付けるサブステップと、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数の前

記位置の第4のリストに、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中タグを付けるサブステップと、

現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に現在のビットプレーン用の位置の前記第3のリストを生成するサブステップであって、前記第3のリストが、現在のビットプレーン用の位置の前記第2のリスト内の位置と、前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されたシンボルを持つ係数の位置と、を除くコード・ブロックの全ての位置リストとを含むように構成されるサブステップと、

現在のビットプレーンの前記クリーンアップ・パス中エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンのクリーンアップ・パス中に1の有意状態に変化する係数である、コード・ブロック内の前記係数の位置の第5のリストを、現在のビットプレーンのクリーンアップ・パス中に生成するサブステップと、現在のビットプレーンのクリーンアップ・パス中にエントロピ符号化されたシンボルを持ち、現在のビットプレーンのクリーンアップ・パス中にゼロの有意状態を持つ係数である、コード・ブロック内の前記係数の位置の第6のリストを、現在のビットプレーンの前記クリーンアップ・パス中に生成するサブステップと、

現在のビットプレーンの前記クリーンアップ・パス中に前記タグを付けた第1の位置リストと、タグを付けなかった第1の位置リストとを現在のビットプレーンについてソートして、それぞれ第7の位置リストと、第8の位置リストとに変えるサブステップと、

現在のビットプレーンの前記クリーンアップ・パス中に前記タグを付けた第4の位置リストと、タグを付けなかった第4の位置リストとを現在のビットプレーンについてソートして、それぞれ第9の位置リストと、第10の位置リストとに変えるサブステップと、

現在のビットプレーンのクリーンアップ・パス中に前記第2、第5、第7及び第9のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第2のリストを生成するサブステップと、

現在のビットプレーンのクリーンアップ・パス中に前記第6、第8及び第10のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストを生成するサブステップと、

現在のビットプレーンのクリーンアップ・パス中に次のビットプレーンのための前記生成された第1のリストから、1の有意状態を持つ係数である、コード・ブロック内の任意の係数の位置を取り除くサブステップと、を有することを特徴とする方法。

【請求項37】 デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化する装

置において、

現在のビットプレーン内でエントロピ符号化されたシンボルをそれぞれ持つ、第1と、第2と、第3の係数の位置リストに従って、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中にコード・ブロックの現在のビットプレーンの前記変換係数の前記シンボルを符号化するエントロピ符号器と、

現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストと、現在のビットプレーンの後の次のビットプレーン用の位置の第2のリストと、現在のビットプレーン用の位置の第3のリストとを生成するリスト・メモリ・マネージャと、を有し、

前記リスト・メモリ・マネージャは、

ゼロの非有意状態を持つ係数であって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されるシンボルである、1の有意状態を持つシンボルを持つ係数に隣接する前記係数である、前記コード・ブロック内の前記係数の位置の第4のリストを、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中生成する手段と、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数である、現在のビットプレーン内の位置の係数の第1のリストの位置に、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中タグを付ける手段と、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数の前記位置の第4のリストに、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中タグを付ける手段と、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に現在のビットプレーン用の位置の前記第3のリストを生成する手段であって、前記第3のリストが、現在のビットプレーン用の位置の前記第2のリスト内の位置と、前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されたシンボルを持つ係数の位置と、を除くコード・ブロックのすべての位置リストとを含むように構成される手段と、

現在のビットプレーンの前記クリーンアップ・パス中エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンの前記クリーンアップ・パス中に1の有意状態に変化する係数である、前記コード・ブロック内の前記係数の位置の第5のリストを、現在のビットプレーンの前記クリーンアップ・パス中生成する手段と、

現在のビットプレーンの前記クリーンアップ・パス中エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンの前記クリーンアップ・パス中ゼロの有意状態を持つ係数である、前記コード・ブロック内

の前記係数の位置の第6のリストを、現在のビットプレーンの前記クリーンアップ・パス中生成する手段と、現在のビットプレーンの前記クリーンアップ・パス中に前記タグを付けた第1の位置リストと、タグを付けなかった第1の位置リストとを現在のビットプレーンについてソートして、それぞれ第7の位置リストと、第8の位置リストとに変える手段と、現在のビットプレーンの前記クリーンアップ・パス中に前記タグを付けた第4の位置リストと、タグを付けなかった第4の位置リストとを現在のビットプレーンについてソートして、それぞれ第9の位置リストと、第10の位置リストとに変える手段と、現在のビットプレーンの前記クリーンアップ・パス中に前記第2、第5、第7及び第9のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第2のリストを生成する手段と、現在のビットプレーンの前記クリーンアップ・パス中に前記第6、第8及び第10のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストを生成する手段と、現在のビットプレーンの前記クリーンアップ・パス中に次のビットプレーンのための前記生成された第1のリストから、1の有意状態を持つ係数である、前記コード・ブロック内の任意の係数の位置を取り除く手段と、を有することを特徴とする装置。

【請求項38】 デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化するコンピュータ・プログラムを有するコンピュータ・プログラム製品であって、

前記コンピュータ・プログラムは、

現在のビットプレーン内でそれぞれエントロピ符号化されたシンボルを持つ係数の第1と、第2と、第3の位置リストに従って、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中にコード・ブロックの現在のビットプレーンの前記変換係数の前記シンボルをエントロピ符号化するためのコードと、

現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストと、現在のビットプレーンの後の次のビットプレーン用の位置の第2のリストと、現在のビットプレーン用の位置の第3のリストとを生成するためのコードと、を有し、

前記生成するための用コードは、

ゼロの非有意状態を持つ係数であって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されるシンボルである、1の有意状態を持つシンボルを持つ係数に隣接する前記係数である、前記コード・ブロック内の前記係数の位置の第4のリストを、現在のビットプレーンの前記シグニフィカン

ス・プロパゲーション・パス中生成するコードと、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数である、現在のビットプレーン内の係数の位置の第1のリストの位置に、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中タグを付けるコードと、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数の前記位置の第4のリストの位置に対して、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中タグを付けるコードと、

現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中に現在のビットプレーン用の位置の前記第3のリストを生成するコードであって、前記第3のリストが、現在のビットプレーン用の位置の前記第2のリスト内の位置と、前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されたシンボルを持つ係数の位置と、を除くコード・ブロックの全ての位置リストとを含むように構成されるコードと、

現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持ち、さらに、現在のビットプレーンのクリーンアップ・パス中に1の有意状態に変化するコード・ブロック内の前記係数の位置の第5のリストを、現在のビットプレーンの前記クリーンアップ・パス中生成するコードと、

現在のビットプレーンの前記クリーンアップ・パス中エントロピ符号化されたシンボルを持ち、現在のビットプレーンの前記クリーンアップ・パス中ゼロの有意状態を持つ係数である、コード・ブロック内の前記係数の位置の第6のリストを、現在のビットプレーンの前記クリーンアップ・パス中に生成するコードと、

現在のビットプレーンの前記クリーンアップ・パス中に前記タグを付けた第1の位置リストと、タグを付けなかった第1の位置リストとを現在のビットプレーンについてソートして、それぞれ第7の位置リストと、第8の位置リストとに変えるコードと、

現在のビットプレーンの前記クリーンアップ・パス中に前記タグを付けた第4の位置リストと、タグを付けなかった第4の位置リストとを現在のビットプレーンについてソートして、それぞれ第9の位置リストと、第10の位置リストとに変えるコードと、

現在のビットプレーンの前記クリーンアップ・パス中に前記第2、第5、第7及び第9のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第2のリストを生成するコードと、

現在のビットプレーンの前記クリーンアップ・パス中に前記第6、第8及び第10のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプ

レーン用の位置の前記第1のリストを生成するコードと、現在のビットプレーンの前記クリーンアップ・パス中に次のビットプレーンのための前記生成された第1のリストから、1の有意状態を持つ係数である、前記コード・ブロック内の任意の係数の位置を取り除くコードと、を有することを特徴とするコンピュータ・プログラム製品。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は一般にデータ圧縮と解凍に関し、特に、エントロピ符号化及び復号化方法及びその装置に関する。本発明はまた、エントロピ符号化を行うためのコンピュータ・プログラムを有するコンピュータ・プログラム製品に関するものである。

【0002】

【従来の技術】新しいJPEG-2000規格に対する提案要求が最近出され、“情報技術—JPEG 2000画像符号化システム—JPEG 2000委員会草案バージョン1.0（1999年12月9日）という表題の規格草案（本明細書では以後JPEG 2000と呼ぶ）が発表された。

【0003】JPEG 2000では1以上の画像タイル成分に画像全体を分割し、次いで、この画像タイル成分の各々の2次元離散ウェーブレット変換を行うことが提案されている。次いで、各画像タイル成分の変換係数がサブバンドにグループ化される。次いで、各コード・ブロックがエントロピ符号化される前に、該サブバンドは分割されて矩形のコード・ブロックに変えられる。各コード・ブロックの変換係数はエントロピ符号化に先行して符号絶対値で表現される。エントロピ符号化は、コンテキスト生成部と算術符号化部の2つの部分から構成される。算術符号化部は、符号化の対象となる係数のビット・シンボルと、そのビット・シンボルのコンテキストとを入力としている。ある係数のビット・シンボルのコンテキストは、コード・ブロックの同じビットプレーン内の8個の包囲係数の‘有意（significance）’状態に基づいている。該ビット・シンボルは算術符号化部によって符号化される。係数の‘有意’状態とは、2進値変数 $Si[m, n]$ であり、該2進値変数は0に初期化されるが、係数の第1のノンゼロ・ビット値が符号化された直後にこの変数は1に変化する。図1は、係数“ $X[m, n]$ ”の8個の包囲係数の近傍有意状態 $Si[m-1, n-1]$ 、 $Si[m-1, n]$ 、 $Si[m-1, n+1]$ 、 $Si[m, n-1]$ 、 $Si[m, n+1]$ 、 $Si[m+1, n-1]$ 、 $Si[m+1, n]$ と $Si[m+1, n+1]$ を示す。但し、 m, n はそれぞれコード・ブロックの行番号と列番号であり、 $Si[i]$ は、係数 $X[m, n]$ のビット・シンボル i が符号化される直前の有意状態である。これらの近傍有意状態は係数 $X[m, n]$ の 3×3 近

傍有意状態と呼ばれることもある。

【0004】算術符号化部 $[m, n]$ は、1つのコード・ブロックの最上位ビットプレーンの全てのビット・シンボルをまず符号化し、次いで、コード・ブロックの次に低いビットプレーンの全てのビット・シンボル等々を符号化し、最後に最下位ビットプレーンの全てのビット・シンボルを符号化する。1つのコード・ブロックの各ビットプレーンの範囲内で、算術符号化部は所定の順序の3つのパスで係数のビット・シンボルを符号化する。

【0005】ビットプレーンの第1のパスはシグニフィカンス・プロパゲーション・パス（SPパス）と呼ばれ、ビットプレーンの第2のパスはマグニチュード・リファインメント・パス（MRパス）と呼ばれ、ビットプレーンの第3で最後のパスはクリーンアップ・パス（Nパス）と呼ばれる。SPパス中で、符号化対象ビット・シンボルは1の有意状態を持っているが、該ビット・シンボル自身が0の有意状態を持つ近傍ビット・シンボルを持っている場合、ビットプレーンのビット・シンボルは符号化される。MRパス中で、ビットプレーンのビット・シンボルがまだ符号化されていない場合、該ビット・シンボルの係数が前回の符号化ビットプレーンですでに有意であれば、該ビット・シンボルは符号化される。Nパス中で、予め符号化されなかったビットプレーンの残りのビット・シンボルはこの時符号化される。

【0006】このコンテキストは符号化対象ビットと共に算術符号化部へ転送され、符号化されたシンボルはビット・ストリームへ出力される。符号化対象のこのビット・シンボルの値が1で、かつ、その有意状態がゼロであれば、いったんビット・シンボルが符号化され、かつ、直後の符号化対象ビット・シンボルが係数用の符号ビットであった場合、有意状態は1にセットされる。上記条件を満たさない場合には有意状態はゼロ（0）のままである。連続する係数とパスのコンテキストを考慮する場合、この係数の最も現在の有意状態が用いられる。

【0007】算術符号化部は、所定の同じ順序で3つのパス（SP、MR、N）内のビットプレーンのビット・シンボルを符号化する。算術符号化部はまず中にノンゼロ・ビットを持つ最上位ビットプレーンへ進み、SP、MRパスをスキップし、Nパスから始まる。次いで、算術符号化部は次に低いビットプレーンへ進み、3つのパス（SP、MR、N）で、その順序でビット・シンボルを符号化する。次いで、算術符号化部は次に低いビットプレーンへ進み、同順のパス（SP、MR、N）でビット・シンボル等々を符号化し、最後に最下位ビットプレーンを符号化する。

【0008】さらに、コード・ブロックの各ビットプレーンは特別の順序で走査される。最上部左側から始まり、列の最初の4個のビット・シンボルが走査される。次いで、コード・ブロックの幅がカバーされてしまうまで、第2列の最初の4個のビット・シンボルが走査され

る。次いで、第1列の第2番目の4個のビット・シンボル等々と走査される。任意の残りの行について、サブバンドの最下位コード・ブロック内で同様の走査が続けられる。図2は、8×8ブロックで構成される64個の変換係数を持つコード・ブロックについてのコード・ブロック走査パターンの一例を示す。この例でわかるように、走査は4つの行の連続するストリップ（細長い片）で行われる。コード・ブロックは8×8ブロックに限定されるものではなく、64×64のコード・ブロック等のさらに大きなコード・ブロックが考えられる。後者の場合、4つの行の16個の連続するストリップが生じる。説明を簡単にするために、本明細書ではこの走査順をJPEG2000走査順と呼ぶことにする。

【0009】JPEG2000に記述されているエントロピ復号化は、エントロピ符号化と鏡像の関係になる。例えば、デコードは符号化の場合と同順でシンボルを復号する。エントロピ復号化はまた、コンテキスト生成部と算術復号部の2つのセクションから構成される。算術復号化部は、復号の対象となるシンボルと、復号の対象となるそのシンボルのコンテキストとを入力として受け取る。復号の対象となる走査順序シンボルのコンテキストは、コード・ブロックの同じビットプレーン内の8個の包囲係数の‘有意’状態に基づき、該シンボルは算術復号化部によって復号される。係数の‘有意’状態は2進値変数 $S[m, n]$ であり、該2進値変数は0に初期化されるが、係数の第1のノンゼロ・ビットプレーン値が復号されると1に変化する。このように、復号化段階中の係数の有意状態は符号化段階中の係数の有意状態と鏡像関係になる（図2参照）。

【0010】JPEG2000では、算術符号化と復号化とは最上位ビットプレーンから最下位ビットプレーンへビットプレーン毎に行われる。初めに、符号化／復号化は、その中にゼロしか含まないすべてのビットプレーンをスキップし、その中にノンゼロ要素を持つ第1のビットプレーンに対するオペレーションを開始する。いったん符号化／復号化が、その中にノンゼロ要素を持つ第1のビットプレーンに達すると、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パスの3つのパスで、コーデックは各連続ビットプレーンに対して作用する。

【0011】ビットプレーン全体にわたるコーデック走査を有し、或る特徴を持つ位置の配置を定めるJPEG2000を実現するエントロピ・コーデックが提案されている。例えば、シグニフィカンス・プロパゲーション・パスで、コーデックは、位置自体は有意でないが、その位置の3×3の近傍に少なくとも1つの有意係数を持つ位置を探す。マグニチュード・リファインメント・パスで、コーデックは既に有意な位置を探す。クリーンアップ・パスで、コーデックは前に符号化／復号化されて

いない位置を探す。したがって、この提案されたコーデックでは、全てのビットプレーンについて3回の走査が行われる。これは、最大スループットが3サイクル当たり1シンボルであることを含意する。

【0012】上記から解るように、提案されたコーデックの主要な問題点として、各パスについて、コーデックが全ての位置を走査し、現在のパスで符号化／復号化が行われているかどうかを判定する必要があるという点が挙げられる。これは、符号化の必要がない位置を走査してサイクルが浪費されることを意味する。

【0013】

【発明が解決しようとする課題】本構成及び／又は提案の1以上の問題点を実質的に解決または少なくとも改善することが本発明の目的である。

【0014】

【課題を解決するための手段】本発明の第1の態様によれば、デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化する方法が提供される。該方法は第1のビットプレーンから所定の最下位ビットプレーンまでの各ビットプレーンについて以下のステップ：現在のビットプレーン内にゼロの有意状態を持つ係数であって、現在のビットプレーン内に1の有意状態を持つ近傍係数を持つ係数に属するシンボルである、現在のビットプレーン内のすべてのシンボルのエントロピ符号化を行うシグニフィカンス・プロパゲーション・ステップと、；前回のビットプレーン内に1の有意状態を持つ係数に属するシンボルである、現在のビットプレーン内のすべてのシンボルのエントロピ符号化を行うマグニチュード・リファインメント・ステップと、；現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップまたはマグニチュード・リファインメント・ステップ中、それ以前にエントロピ符号化されなかったシンボルである、現在のビットプレーン内のすべてのシンボルのエントロピ符号化を行うクリーンアップ・ステップと、を実行する方法において、現在のビットプレーンのシグニフィカンス・プロパゲーション・ステップ中エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第1のリストを、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップに先行して生成するステップと、；現在のビットプレーンのマグニチュード・リファインメント・ステップ中エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第2のリストを、現在のビットプレーンのマグニチュード・リファインメント・ステップに先行して生成するステップと、；現在のビットプレーンの前記クリーンアップ・ステップ中エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第3のリストを、現在のビットプレーンのクリーンアップ・ステッ

とを生成するステップと、が提供される。この場合、該生成ステップは、ゼロの非有意状態を持つ係数であって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されるシンボルである、1の有意状態を持つシンボルを持つ係数に隣接する前記係数である、前記コード・ブロック内の前記係数の位置の第4のリストを現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中生成するサブステップと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数である、現在のビットプレーン内の係数の位置の第1のリストの位置に現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付けるサブステップと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数の前記位置の第4のリストに現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付けるサブステップと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に現在のビットプレーン用の位置の前記第3のリストを生成するサブステップであって、前記第3のリストが、現在のビットプレーン用の位置の前記第2のリスト内の位置と、シグニフィカンス・プロパゲーション・パス中エントロピ符号化されたシンボルを持つ係数の位置と、を除くコード・ブロックのすべての位置リストとを含むように構成されるサブステップと、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンのクリーンアップ・パス中に1の有意状態に変化する係数である、コード・ブロック内の前記係数の位置の第5のリストを現在のビットプレーンのクリーンアップ・パス中生成するサブステップと、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持ち、現在のビットプレーンのクリーンアップ・パス中ゼロの有意状態を持つ係数である、コード・ブロック内の前記係数の位置の第6のリストを現在のビットプレーンのクリーンアップ・パス中生成するサブステップと、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第1の位置リストと、タグを付けなかった第1の位置リストとを現在のビットプレーンについてソートして、それぞれ第7の位置リストと、第8の位置リストとに変えるサブステップと、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第4の位置リストと、タグを付けなかった第4の位置リストとを現在のビットプレーンについてソートして、それぞれ第9の位置リストと、第10の位置リストとに変えるサブステップと、現在のビットプレーンのクリーンアップ・パス中に前記第2、第5、第7及び第9のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーンの位置の前記第2のリストを生成するサブステップ

と、現在のビットプレーンのクリーンアップ・パス中に前記第6、第8及び第10のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーンの位置の前記第1のリストを生成するサブステップと、現在のビットプレーンのクリーンアップ・パス中に次のビットプレーンのための前記生成された第1のリストから、1の有意状態を持つ係数である、コード・ブロック内の任意の係数の位置を取り除くサブステップと、を有する。

【0021】本発明の別の態様によれば、デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化する装置が提供される。該装置は、現在のビットプレーン内でエントロピ符号化されたシンボルをそれぞれ持つ、第1と、第2と、第3の係数の位置リストに従って、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中にコード・ブロックの現在のビットプレーンの前記変換係数の前記シンボルを符号化するエントロピ符号器と、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストと、現在のビットプレーンの後の次のビットプレーン用の位置の第2のリストと、現在のビットプレーン用の位置の第3のリストとを生成するリスト・メモリ・マネージャと、を有し、リスト・メモリ・マネージャが、ゼロの非有意状態を持つ係数であって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されるシンボルである、1の有意状態を持つシンボルを持つ係数に隣接する前記係数である、前記コード・ブロック内の前記係数の位置の第4のリストを現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中生成する手段と、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数である、現在のビットプレーン内の係数の位置の第1のリストの位置に現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付ける手段と、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数の前記位置の第4のリストに現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付ける手段と、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に現在のビットプレーン用の位置の前記第3のリストを生成する手段であって、前記第3のリストが、現在のビットプレーン用の位置の前記第2のリスト内の位置と、シグニフィカンス・プロパゲーション・パス中エントロピ符号化されたシンボルを持つ係数の位置と、を除くコード・ブロックのすべての位置リストとを含むように構成される手段と、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンのクリーンアップ・パス中に1の有

る。次いで、第1列の第2番目の4個のビット・シンボル等々と走査される。任意の残りの行について、サブバンドの最下位コード・ブロック内で同様の走査が続けられる。図2は、8×8ブロックで構成される64個の変換係数を持つコード・ブロックについてのコード・ブロック走査パターンの一例を示す。この例でわかるように、走査は4つの行の連続するストリップ（細長い片）で行われる。コード・ブロックは8×8ブロックに限定されるものではなく、64×64のコード・ブロック等のさらに大きなコード・ブロックが考えられる。後者の場合、4つの行の16個の連続するストリップが生じる。説明を簡単にするために、本明細書ではこの走査順をJPEG2000走査順と呼ぶことにする。

【0009】JPEG2000に記述されているエントロピ復号化は、エントロピ符号化と鏡像の関係になる。例えば、デコーダは符号化の場合と同順でシンボルを復号する。エントロピ復号化はまた、コンテキスト生成部と算術復号部の2つのセクションから構成される。算術復号化部は、復号の対象となるシンボルと、復号の対象となるそのシンボルのコンテキストとを入力として受け取る。復号の対象となる走査順序シンボルのコンテキストは、コード・ブロックの同じビットプレーン内の8個の包囲係数の‘有意’状態に基づき、該シンボルは算術復号化部によって復号される。係数の‘有意’状態は2進値変数 $S[m, n]$ であり、該2進値変数は0に初期化されるが、係数の第1のノンゼロ・ビットプレーン値が復号されると1に変化する。このように、復号化段階中の係数の有意状態は符号化段階中の係数の有意状態と鏡像関係になる（図2参照）。

【0010】JPEG2000では、算術符号化と復号化とは最上位ビットプレーンから最下位ビットプレーンへビットプレーン毎に行われる。初めに、符号化／復号化は、その中にゼロしか含まないすべてのビットプレーンをスキップし、その中にノンゼロ要素を持つ第1のビットプレーンに対するオペレーションを開始する。いったん符号化／復号化が、その中にノンゼロ要素を持つ第1のビットプレーンに達すると、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パスの3つのパスで、コーデックは各連続ビットプレーンに対して作用する。

【0011】ビットプレーン全体にわたるコーデック走査を有し、或る特徴を持つ位置の配置を定めるJPEG2000を実現するエントロピ・コーデックが提案されている。例えば、シグニフィカンス・プロパゲーション・パスで、コーデックは、位置自体は有意でないが、その位置の3×3の近傍に少なくとも1つの有意係数を持つ位置を探す。マグニチュード・リファインメント・パスで、コーデックは既に有意な位置を探す。クリーンアップ・パスで、コーデックは前に符号化／復号化されて

いない位置を探す。したがって、この提案されたコーデックでは、全てのビットプレーンについて3回の走査が行われる。これは、最大スループットが3サイクル当たり1シンボルであることを含意する。

【0012】上記から解るように、提案されたコーデックの主要な問題点として、各パスについて、コーデックが全ての位置を走査し、現在のパスで符号化／復号化が行われているかどうかを判定する必要があるという点が挙げられる。これは、符号化の必要がない位置を走査してサイクルが浪費されることを意味する。

【0013】

【発明が解決しようとする課題】本構成及び／又は提案の1以上の問題点を実質的に解決または少なくとも改善することが本発明の目的である。

【0014】

【課題を解決するための手段】本発明の第1の態様によれば、デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化する方法が提供される。該方法は第1のビットプレーンから所定の最下位ビットプレーンまでの各ビットプレーンについて以下のステップ：現在のビットプレーン内にゼロの有意状態を持つ係数であって、現在のビットプレーン内に1の有意状態を持つ近傍係数を持つ係数に属するシンボルである、現在のビットプレーン内のすべてのシンボルのエントロピ符号化を行うシグニフィカンス・プロパゲーション・ステップと、；前回のビットプレーン内に1の有意状態を持つ係数に属するシンボルである、現在のビットプレーン内のすべてのシンボルのエントロピ符号化を行うマグニチュード・リファインメント・ステップと、；現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップまたはマグニチュード・リファインメント・ステップ中、それ以前にエントロピ符号化されなかったシンボルである、現在のビットプレーン内のすべてのシンボルのエントロピ符号化を行うグリーンアップ・ステップと、を実行する方法において、現在のビットプレーンのシグニフィカンス・プロパゲーション・ステップ中エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第1のリストを、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・ステップに先行して生成するステップと、；現在のビットプレーンのマグニチュード・リファインメント・ステップ中エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第2のリストを、現在のビットプレーンのマグニチュード・リファインメント・ステップに先行して生成するステップと、；現在のビットプレーンの前記グリーンアップ・ステップ中エントロピ符号化の対象となるシンボルを持つ係数である、前記コード・ブロック内の前記係数の位置の第3のリストを、現在のビットプレーンのグリーンアップ・ステッ

ブに先行して生成するステップとをさらに有する。

【0015】本発明の別の態様によれば、デジタル画像の変換係数を有する、コード・ブロックを表すシンボルをエントロピ符号化する装置が提供される。該装置は、第1のビットプレーンから所定の最下位ビットプレーンまで前記コード・ブロックをシンボルから成るビットプレーンに分割するビットプレーン・スプリッタと；シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中に前記シンボルを符号化するエントロピ符号器と；現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中エントロピ符号化の対象となるシンボルを持つ係数である、コード・ブロック内の前記係数の位置の第1のリストを、現在のビットプレーンのシグニフィカンス・プロパゲーション・パスに先行して生成し、現在のビットプレーンのマグニチュード・リファインメント・パス中エントロピ符号化の対象となるシンボルを持つ係数である、コード・ブロック内の前記係数の位置の第2のリストを、現在のビットプレーンのマグニチュード・リファインメント・パスに先行して生成し、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化の対象となるシンボルを持つ係数である、コード・ブロック内の前記係数の位置の第3のリストを、現在のビットプレーンのクリーンアップ・パスに先行して生成するリスト・メモリ・マネージャと；前記第1のリストと、第2のリストと、第3の位置リストとを格納する記憶装置と、を有する。

【0016】本発明の別の態様によれば、デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化するコンピュータ・プログラムを有するコンピュータ・プログラム製品が提供される。該コンピュータ・プログラムは、第1のビットプレーンから所定の最下位ビットプレーンへシンボルから成るビットプレーンに前記コード・ブロックを分割するためのコードと、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中に前記シンボルをエントロピ符号化するためのコードと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中エントロピ符号化の対象となるシンボルを持つ係数である、コード・ブロック内の前記係数の位置の第1のリストを、現在のビットプレーンのシグニフィカンス・プロパゲーション・パスに先行して生成するためのコードと、現在のビットプレーンのマグニチュード・リファインメント・パス中エントロピ符号化の対象となるシンボルを持つ係数である、コード・ブロック内の前記係数の位置の第2のリストを、現在のビットプレーンのマグニチュード・リファインメント・パスに先行して生成するためのコードと、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化の対象となるシンボルを持つ係数である、コード・ブ

ロック内の前記係数の位置の第3のリストを、現在のビットプレーンのクリーンアップ・パスに先行して生成するためのコードと、前記第1の位置リストと、第2の位置リストと、第3の位置リストとを格納するためのコードと、を有する。

【0017】本発明の別の態様によれば、係数を有するブロックを表すエントロピ符号化シンボルの方法が提供される。該方法は、第1のビットプレーンから所定の最下位ビットプレーンまでの各ビットプレーンについて以下のステップ：ブロックの現在のビットプレーン中エントロピ符号化されるシンボルを有する係数である、ブロック内の前記係数の位置の少なくとも1つのリストを生成するステップと、前記少なくとも1つのリストを用いて、そのブロックの現在のビットプレーン内の係数の前記シンボルをエントロピ符号化するステップと、を実行する。

【0018】本発明の別の態様によれば、係数を有するブロックを表すシンボルをエントロピ符号化する装置が提供される。該装置は、ブロックの現在のビットプレーン中エントロピ符号化されるシンボルを持つ係数である、ブロック内の前記係数の位置の少なくとも1つのリストを生成する生成装置と、前記少なくとも1つのリストを用いて、そのブロックの現在のビットプレーン内の係数の前記シンボルをエントロピ符号化するエントロピ符号器と、を有する。

【0019】本発明の別の態様によれば、係数を有するブロックを表すシンボルをエントロピ符号化するコンピュータ・プログラムを有するコンピュータ・プログラム製品が提供される。該コンピュータ・プログラムは、ブロックの現在のビットプレーン中エントロピ符号化されるシンボルを有する係数である、ブロック内の前記係数の位置の少なくとも1つのリストを生成するためのコードと、前記少なくとも1つのリストを用いて、そのブロックの現在のビットプレーン内の係数の前記シンボルをエントロピ符号化するためのコードと、を有する。

【0020】本発明の別の態様によれば、デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化する方法が提供される。該方法は、第1のビットプレーンから所定の最下位ビットプレーンまでの各ビットプレーンについて以下のステップ：現在のビットプレーン内でエントロピ符号化の対象となる前記シンボルをそれぞれ持つ係数の、第1と、第2と、第3の位置リストに従って、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中にコード・ブロックの現在のビットプレーンの前記変換係数のシンボルを符号化するステップと、現在のビットプレーンの後の次のビットプレーン用の位置の第1のリストと、現在のビットプレーンの後の次のビットプレーン用の位置の第2のリストと、現在のビットプレーン用の位置の第3のリスト

とを生成するステップと、が提供される。この場合、該生成ステップは、ゼロの非有意状態を持つ係数であって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されるシンボルである、1の有意状態を持つシンボルを持つ係数に隣接する前記係数である、前記コード・ブロック内の前記係数の位置の第4のリストを現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中生成するサブステップと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数である、現在のビットプレーン内の係数の位置の第1のリストの位置に現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付けるサブステップと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数の前記位置の第4のリストに現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付けるサブステップと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に現在のビットプレーン用の位置の前記第3のリストを生成するサブステップであって、前記第3のリストが、現在のビットプレーン用の位置の前記第2のリスト内の位置と、シグニフィカンス・プロパゲーション・パス中エントロピ符号化されたシンボルを持つ係数の位置と、を除くコード・ブロックのすべての位置リストとを含むように構成されるサブステップと、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンのクリーンアップ・パス中に1の有意状態に変化する係数である、コード・ブロック内の前記係数の位置の第5のリストを現在のビットプレーンのクリーンアップ・パス中生成するサブステップと、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持ち、現在のビットプレーンのクリーンアップ・パス中ゼロの有意状態を持つ係数である、コード・ブロック内の前記係数の位置の第6のリストを現在のビットプレーンのクリーンアップ・パス中生成するサブステップと、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第1の位置リストと、タグを付けた第1の位置リストとを現在のビットプレーンについてソートして、それぞれ第7の位置リストと、第8の位置リストとに変えるサブステップと、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第4の位置リストと、タグを付けた第4の位置リストとを現在のビットプレーンについてソートして、それぞれ第9の位置リストと、第10の位置リストとに変えるサブステップと、現在のビットプレーンのクリーンアップ・パス中に前記第2、第5、第7及び第9のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーンの位置の前記第2のリストを生成するサブステップ

と、現在のビットプレーンのクリーンアップ・パス中に前記第6、第8及び第10のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーンの位置の前記第1のリストを生成するサブステップと、現在のビットプレーンのクリーンアップ・パス中に次のビットプレーンのための前記生成された第1のリストから、1の有意状態を持つ係数である、コード・ブロック内の任意の係数の位置を取り除くサブステップと、を有する。

【0021】本発明の別の態様によれば、デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化する装置が提供される。該装置は、現在のビットプレーン内でエントロピ符号化されたシンボルをそれぞれ持つ、第1と、第2と、第3の係数の位置リストに従って、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中にコード・ブロックの現在のビットプレーンの前記変換係数の前記シンボルを符号化するエントロピ符号器と、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストと、現在のビットプレーンの後の次のビットプレーン用の位置の第2のリストと、現在のビットプレーン用の位置の第3のリストとを生成するリスト・メモリ・マネージャと、を有し、リスト・メモリ・マネージャが、ゼロの非有意状態を持つ係数であって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されるシンボルである、1の有意状態を持つシンボルを持つ係数に隣接する前記係数である、前記コード・ブロック内の前記係数の位置の第4のリストを現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中生成する手段と、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数である、現在のビットプレーン内の係数の位置の第1のリストの位置に現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付ける手段と、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数の前記位置の第4のリストに現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付ける手段と、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に現在のビットプレーン用の位置の前記第3のリストを生成する手段であって、前記第3のリストが、現在のビットプレーン用の位置の前記第2のリスト内の位置と、シグニフィカンス・プロパゲーション・パス中エントロピ符号化されたシンボルを持つ係数の位置と、を除くコード・ブロックのすべての位置リストとを含むように構成される手段と、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンのクリーンアップ・パス中に1の有

意状態に変化する係数である、コード・ブロック内の前記係数の位置の第5のリストを現在のビットプレーンのクリーンアップ・パス中生成する手段と、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンのクリーンアップ・パス中ゼロの有意状態を持つ係数である、コード・ブロック内の前記係数の位置の第6のリストを現在のビットプレーンのクリーンアップ・パス中生成する手段と、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第1の位置リストと、タグを付けなかった第1の位置リストとを現在のビットプレーンについてソートして、それぞれ第7の位置リストと、第8の位置リストとに変える手段と、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第4の位置リストと、タグを付けなかった第4の位置リストとを現在のビットプレーンについてソートして、それぞれ第9の位置リストと、第10の位置リストとに変える手段と、現在のビットプレーンのクリーンアップ・パス中に前記第2、第5、第7及び第9のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第2のリストを生成する手段と、現在のビットプレーンのクリーンアップ・パス中に前記第6、第8及び第10のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストを生成する手段と、現在のビットプレーンのクリーンアップ・パス中に次のビットプレーンのための前記生成された第1のリストから、1の有意状態を持つ係数である、コード・ブロック内の任意の係数の位置を取り除く手段と、を有する。

【0022】本発明の別の態様によれば、デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化するコンピュータ・プログラムを有するコンピュータ・プログラム製品が提供される。該コンピュータ・プログラムは、現在のビットプレーン内でそれぞれエントロピ符号化されたシンボルを持つ係数の第1と、第2と、第3の位置リストに従って、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中にコード・ブロックの現在のビットプレーンの前記変換係数の前記シンボルをエントロピ符号化するためのコードと、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストと、現在のビットプレーンの後の次のビットプレーン用の位置の第2のリストと、現在のビットプレーン用の位置の第3のリストとを生成するためのコードと、を有し、その場合、該生成コードは、ゼロの非有意状態を持つ係数であって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されるシンボルである、1の有意状態を持つシンボルを持つ係数に隣接する前記係数

である、前記コード・ブロック内の前記係数の位置の第4のリストを現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中生成するコードと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数である、現在のビットプレーン内の係数の位置の第1のリストの位置に現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付けるコードと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数の前記位置の第4のリストの位置に対して現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付けるコードと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に現在のビットプレーン用の位置の前記第3のリストを生成するコードであって、前記第3のリストが、現在のビットプレーン用の位置の前記第2のリスト内の位置と、シグニフィカンス・プロパゲーション・パス中エントロピ符号化されたシンボルを持つ係数の位置と、を除くコード・ブロックのすべての位置リストとを含むように構成されるコードと、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持ち、さらに、現在のビットプレーンのクリーンアップ・パス中に1の有意状態に変化するコード・ブロック内の前記係数の位置の第5のリストを現在のビットプレーンのクリーンアップ・パス中生成するコードと、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持ち、現在のビットプレーンのクリーンアップ・パス中ゼロの有意状態を持つ係数である、コード・ブロック内の前記係数の位置の第6のリストを現在のビットプレーンのクリーンアップ・パス中生成するコードと、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第1の位置リストと、タグを付けなかった第1の位置リストとを現在のビットプレーンについてソートして、それぞれ第7の位置リストと、第8の位置リストとに変えるコードと、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第4の位置リストと、タグを付けなかった第4の位置リストとを現在のビットプレーンについてソートして、それぞれ第9の位置リストと、第10の位置リストとに変えるコードと、現在のビットプレーンのクリーンアップ・パス中に前記第2、第5、第7及び第9のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第2のリストを生成するコードと、現在のビットプレーンのクリーンアップ・パス中に前記第6、第8及び第10のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストを生成するコードと、現在のビットプレーンのクリーンアップ・パス中に次のビットプレーンのための前記生成された第1のリストから、1の有意状態を持

意状態に変化する係数である、コード・ブロック内の前記係数の位置の第5のリストを現在のビットプレーンのクリーンアップ・パス中生成する手段と、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持つ係数であって、現在のビットプレーンのクリーンアップ・パス中ゼロの有意状態を持つ係数である、コード・ブロック内の前記係数の位置の第6のリストを現在のビットプレーンのクリーンアップ・パス中生成する手段と、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第1の位置リストと、タグを付けなかった第1の位置リストとを現在のビットプレーンについてソートして、それぞれ第7の位置リストと、第8の位置リストとに変える手段と、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第4の位置リストと、タグを付けなかった第4の位置リストとを現在のビットプレーンについてソートして、それぞれ第9の位置リストと、第10の位置リストとに変える手段と、現在のビットプレーンのクリーンアップ・パス中に前記第2、第5、第7及び第9のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第2のリストを生成する手段と、現在のビットプレーンのクリーンアップ・パス中に前記第6、第8及び第10のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストを生成する手段と、現在のビットプレーンのクリーンアップ・パス中に次のビットプレーンのための前記生成された第1のリストから、1の有意状態を持つ係数である、コード・ブロック内の任意の係数の位置を取り除く手段と、を有する。

【0022】本発明の別の態様によれば、デジタル画像の変換係数を有するコード・ブロックを表すシンボルをエントロピ符号化するコンピュータ・プログラムを有するコンピュータ・プログラム製品が提供される。該コンピュータ・プログラムは、現在のビットプレーン内でそれぞれエントロピ符号化されたシンボルを持つ係数の第1と、第2と、第3の位置リストに従って、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パス中にコード・ブロックの現在のビットプレーンの前記変換係数の前記シンボルをエントロピ符号化するためのコードと、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストと、現在のビットプレーンの後の次のビットプレーン用の位置の第2のリストと、現在のビットプレーン用の位置の第3のリストとを生成するためのコードと、を有し、その場合、該生成用コードは、ゼロの非有意状態を持つ係数であって、現在のビットプレーンの前記シグニフィカンス・プロパゲーション・パス中エントロピ符号化されるシンボルである、1の有意状態を持つシンボルを持つ係数に隣接する前記係数

である、前記コード・ブロック内の前記係数の位置の第4のリストを現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中生成するコードと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数である、現在のビットプレーン内の係数の位置の第1のリストの位置に現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付けるコードと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に1の有意状態に変化する係数の前記位置の第4のリストの位置に対して現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中タグを付けるコードと、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス中に現在のビットプレーン用の位置の前記第3のリストを生成するコードであって、前記第3のリストが、現在のビットプレーン用の位置の前記第2のリスト内の位置と、シグニフィカンス・プロパゲーション・パス中エントロピ符号化されたシンボルを持つ係数の位置と、を除くコード・ブロックのすべての位置リストとを含むように構成されるコードと、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持ち、さらに、現在のビットプレーンのクリーンアップ・パス中に1の有意状態に変化するコード・ブロック内の前記係数の位置の第5のリストを現在のビットプレーンのクリーンアップ・パス中生成するコードと、現在のビットプレーンのクリーンアップ・パス中エントロピ符号化されたシンボルを持ち、現在のビットプレーンのクリーンアップ・パス中ゼロの有意状態を持つ係数である、コード・ブロック内の前記係数の位置の第6のリストを現在のビットプレーンのクリーンアップ・パス中生成するコードと、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第1の位置リストと、タグを付けなかった第1の位置リストとを現在のビットプレーンについてソートして、それぞれ第7の位置リストと、第8の位置リストとに変えるコードと、現在のビットプレーンのクリーンアップ・パス中に前記タグを付けた第4の位置リストと、タグを付けなかった第4の位置リストとを現在のビットプレーンについてソートして、それぞれ第9の位置リストと、第10の位置リストとに変えるコードと、現在のビットプレーンのクリーンアップ・パス中に前記第2、第5、第7及び第9のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第2のリストを生成するコードと、現在のビットプレーンのクリーンアップ・パス中に前記第6、第8及び第10のリスト上の前記位置を比較して併合し、現在のビットプレーンの後の次のビットプレーン用の位置の前記第1のリストを生成するコードと、現在のビットプレーンのクリーンアップ・パス中に次のビットプレーンのための前記生成された第1のリストから、1の有意状態を持

係数である、コード・ブロック内の任意の係数の位置を取り除くコードと、を有する。

【0023】

【発明の実施の形態】本発明のいくつかの好ましい実施の形態について、以下に図面を参照しながら説明する。

【0024】ステップ及び／又は特徴を示す同じ参照番号を持つ1以上のいずれかの添付図面で参照が行われる場合、それらのステップ及び／又は特徴は、逆の意図が現れない限り、この説明の目的のために同じ機能または作用を持つ。

【0025】本明細書に記載の好ましい装置と方法の原理は、データ圧縮と解凍に対して一般的適用性を持つ。しかし、説明を簡単にするために、上述のJ P E G 2 0 0 0に従うデジタル画像の圧縮と解凍を参照しながら好ましい装置と方法について説明する。しかし、これは記載の方法及び装置に本発明を限定することを意図するものではない。例えば、本発明はデジタル・ビデオの圧縮と解凍に適用することも可能である。

【0026】好ましい方法についての詳細な説明に進む前に、好ましい方法の簡単な概観について以下説明する。まず、画像全体は1以上の画像タイル成分に分割され、次いで、これら画像タイル成分の各々に対して2次元離散ウェーブレット変換が行われる。次いで、各画像タイル成分の変換係数がサブバンドにグループ化される。次いで、該サブバンドは、各コード・ブロックがエントロピ符号化される前にさらに矩形のコード・ブロックに分割される。

【0027】次に図3に目を向けると、第1の構成に従ってコード・ブロックの変換係数のシンボルをエントロピ符号化する方法を示すフローチャート300が示されている。好適には、方法300が画像の各コード・ブロックを呼び出すことが望ましい。本方法はコーデック法で機能する。すなわち、本方法はシンボルを復号もしくは符号化することができる。符号化あるいは復号のいずれかが行われる間、本方法はほぼ同一のステップを実行する。復号処理は、シグニフィカンス・プロパゲーションリストと、マグニチュード・リファインメント・リストと、クリーンアップ・リストとを利用するという点で符号化処理と鏡像関係になる。しかし、復号化中、ビットプレーンに分割されるソースコード・ブロックは存在せず、最上位ビットプレーンのロケーションが既に与えられている。説明を簡単にするために、符号化処理を参照しながら主としてエントロピ符号化法について説明を行う。

【0028】方法300は、ステップ302から始まり、いずれかの必要なパラメータが初期化される。初期化段階中、方法300は有効値テーブルを生成する。この有効値テーブルは符号化の対象となるコード・ブロックの係数の有意状態を示す2次元配列を有する。最初に、配列の各エントリは、各係数が非有意状態を持って

いることを示すゼロ(0)にセットされる。ある係数の第1のノンゼロ・ビットプレーン値が符号化され、配列の対応するエントリが(1)にリセットされたとき、その係数の有意状態は(1)に遷移する。任意の呼び出されたシグニフィカンス・プロパゲーション符号化パス中にまたは、任意のクリーンアップ符号化パス中に有効値テーブルの更新を行うことができる。

【0029】また初期化段階中、本方法によって、本明細書でシグニフィカンス・プロパゲーションリスト(SPリスト)、マグニチュード・リファインメント・リスト(MRリスト)及びクリーンアップ・リスト(Nリスト)と名付けられる3つのリストが生成される。これらのリストには、コード・ブロックの範囲内の、現在のパス中符号化の対象となるビット・シンボルの位置が含まれる。例えば、シグニフィカンス・プロパゲーションリストには、コード・ブロックの範囲内に、現在のシグニフィカンス・プロパゲーション・パス中符号化の対象となるビット・シンボルのすべての位置リストが含まれる。ビット・シンボルの位置という用語は、本明細書では、そのビット・シンボルが一部を形成する変換係数のコード・ブロックの範囲内の配列位置を意味する。さらに、これらのリストの各々の範囲内の位置は、コード・ブロック走査順に従って索引化される(図2参照)。最初、シグニフィカンス・プロパゲーションリスト(SPリスト)と最大リファインメント・リスト(MRリスト)とは空である。一方、クリーンアップ・リスト(Nリスト)はコード・ブロックの範囲内のすべての位置をリストし、ノンゼロ・ビットを持つ最上位のビットプレーンの符号化に使用される。

【0030】SPリストとMRリストとは、必要に応じてさらなる位置の追加と削除により本方法の処理中更新することができる。さらに、現在のNリストは現在のSPとMRリスト内に含まれていない位置を含むようにSP符号化パス中生成される。

【0031】開始302後、コード・ブロックの変換係数が受け取られ、ビットプレーンに分割される(304)。次のステップ306で、コード・ブロックのビットプレーンがチェックされ、どのビットプレーンがその中に最上位ノンゼロ・ビットを持っているかが決定される。

【0032】次いで、本方法はステップ308へ進み、このステップで最上位ビットを持つビットプレーンが単一クリーンアップ・パス内でエントロピ符号化される。具体的には、ノンゼロ・ビットの最上位ビットを持つビットプレーンに属するビット・シンボルが各々エントロピ符号化される。これらのビット・シンボルは、例えば図2に図示のように、所定の走査順でエントロピ符号化される。エントロピ符号化ステップ308は、入力として符号化対象ビット・シンボルとそのコンテキストとをとる。該コンテキストはJ P E G 2 0 0 0に従って好適

に決定される。同時に、シグニフィカンス・プロパゲーション・パスとマグニチュード・リファインメント・パスとの中で処理の対象となる位置のリストが生成され、その結果、本方法は次のビットプレーン内でのビットの効率的処理が可能となる。最初のクリーンアップ・パス308中、ビット・シンボルが有意であることが判明した場合、その関連する位置が現在のMRリストに追加される。最初のクリーンアップ符号化パス中に本方法は、有意であることが判明した位置の 3×3 の近傍内の非有意近傍も決定する。次いで、これらの位置は現在のSPリストに追加される。

【0033】ステップ308の完了後、本方法は判定ブロック310へ進み、そこで処理の対象となるビットプレーンがさらに存在するかどうかのチェックが行われる。もし存在すれば、本方法はステップ312へ進み、そこで次のビットプレーンのビット・シンボルが検索される。

【0034】次いで、本方法は、関連するビット・シンボルを現在のビットプレーンのシグニフィカンス・プロパゲーション・パスで処理するステップ314へ進む。このステップ314で、現在のSPリスト内の位置に対応する現在のビットプレーン内のビット・シンボルがエントロピ符号化される。SP符号化パス314中、ビット・シンボルが有意であることが判明した場合、その関連する位置が新しいMRリストに追加される。2回の符号化を避けるために、次のクリーンアップ符号化パス318の完了後、この新しいMRリストは現在のMRリストと併合される。またSP符号化パス314中、ビット・シンボルが有意であることが判明したとき、本方法はその有効ビット・シンボルの位置(P)の非有意近傍を決定する。次いで、本方法は、走査順でその位置(P)に先行するこれらの非有意近傍の中のいずれの非有意近傍も新しいSPリストに追加する。一方、本方法は、走査順でその位置(P)の後にあるこれらの非有意近傍のいずれの非有意近傍をも現在のSPリストに追加する。次いで、本方法は更新された現在のSPリストの再索引化を行う。SP符号化パス314中、本方法は、更新された現在のリストSPと現在のMRリストとを用いてNリストの生成も開始する。

【0035】SP符号化パス314の完了後、本方法はマグニチュード・リファインメント符号化パス316へ進む。このステップ316で、マグニチュード・リファインメント・リスト内の位置に対応する現在のビットプレーン内のすべてのビット・シンボルがエントロピ符号化される。

【0036】MR符号化パス316の完了後、本方法はクリーンアップ符号化パス318へ進む。クリーンアップ符号化ステップ318で、(前回の符号化パス中生成された)現在のクリーンアップNリスト内の位置に対応する現在のビットプレーン内のすべてのビット・シンボ

ルがエントロピ符号化される。クリーンアップ符号化パス318中、ビット・シンボルが有意であることが判明したとき、その関連する位置は新しいMRリストに追加される。本方法は新しいMRリスト内の位置リストの非有意近傍も決定する。次いで、これらの非有意近傍の位置はクリーンアップ・リストから削除され、新しいSPリストに追加される。次いで、新しいMRリストと新しいSPリストはそれぞれ現在のMRリストとSPリストと併合される。次いで、新しく併合されたMRとSPリストとは走査順に従って再索引化される。

【0037】クリーンアップ符号化パス318の完了後、本方法は任意のさらなるビットプレーン処理を行うために判定ブロック310へ戻る。判定ブロックが真(イエス)を返した場合、すなわち、処理の対象となるさらなるビットプレーンが存在しない場合本方法は終了する(300)。好適には、本方法は、最上位ビットプレーンから所定の最下位ビットプレーンまで有効ビットを含むビットプレーンを処理することが望ましい。符号化段階中、本方法は、符号化されたビット・ストリームのヘッダへ情報を追加する。該情報によって最上位ビットプレーンと所定の最下位ビットプレーンとが特定される。復号段階では、本方法は符号化されたビット・ストリームの復号化時にこのヘッダ情報を使用する。

【0038】次に図4を参照すると、図3に図示のエントロピ符号化法で用いられる最初のクリーンアップ符号化ステップ308のフローチャートが示されている。クリーンアップ符号化ステップ308は、その中に有効ビットを持つ最上位ビットプレーンの処理を開始する。最初のクリーンアップ符号化ステップ308の開始時に、シグニフィカンス・プロパゲーション(SP)リストとマグニチュード・リファインメント(MR)リストは空であり、すべての係数はこのクリーンアップ・パス中符号化される。このビットプレーンの各係数について、最初のクリーンアップ符号化ステップ308は、以下の演算を実行する：(1)係数がこのビットプレーン内で有意である場合、その係数はMRリストに追加される。

(2)その係数の 3×3 の近傍のいずれかの係数が有意であって、しかも、その係数自身が有意でない場合、その係数はSPリストに追加される。

【0039】最初のクリーンアップ符号化ステップ308の最後に、MRリストとSPリストとは完全なものとなり、次のステップ用として使用される準備ができています。最初のクリーンアップ符号化ステップ308は、前処理ステップ412中このビットプレーン内で係数の各ビット・シンボルを符号化することにより、さらに、現在前処理されている(412)係数の有効値に関する情報、および、有意係数の 3×3 の近傍の範囲内の任意の非有意近傍に関する情報をメモリに書き込むことにより、これらの演算を実行する。以下さらに詳細に説明するように、ビット・シンボルが前処理されている間、該

メモリは更新を受ける継続状態にある。クリーンアップ符号化ステップ308は、以下さらに詳細に説明するように、メモリに書き込まれた情報を利用することにより、後処理ステップ418中、係数をMRリストに加えるべきか、それとも、SPリストに加えるべきかについての決定も行う。

【0040】最初のクリーンアップ符号化ステップ308はサブ・プロシージャであり、有効ビットをその中に持つ最上位ビットプレーンを処理する主要な方法300によって呼び出される。最初の符号化ステップ308はステップ402から始まる。次いで、このサブ・プロシージャは変数 i と j が双方とも1にセットされるステップ404と406へ進む。変数 j は列カウンタとして、変数 i はストリップ・カウンタとして使用され、最初のクリーンアップ符号化ステップ308がJPEG2000の走査順でコード・ブロックのビットプレーンの列とストリップを処理することが可能になる。コード・ブロックはビット・シンボルの N 列の M ストリップを持つ。 M と N の値はコード・ブロックのサイズに応じて変動する。

【0041】次いで、最初のクリーンアップ符号化ステップ308は判定ブロック408へ進み、ここで、ストリップ・カウンタ i が1以上か、コード・ブロック内のストリップ M の総数以下であるかどうかのチェックが行われる。判定ブロック408がイエス（真）を返した場合、最初のクリーンアップ符号化ステップ308は判定ブロック410へ進む。次いで、判定ブロック410は、列カウンタ j がコード・ブロック内で1以上であるか、列 N の総数以下であるかをチェックする。判定ブロック410がイエス（真）を返した場合、最初のクリーンアップ符号化ステップ308は前処理ステップ412へ進み、この前処理ステップ412はそのビットプレーンの i 番目のストリップの j 番目の列を前処理し、メモリに情報を書き込む。この前処理ステップ412について図5を参照しながら以下さらに詳細に後述する。前処理ステップ412の完了後、最初のクリーンアップ符号化ステップ308はステップ420へ進む。判定ブロック408と410のうちのいずれか一方がノー（偽）を返した場合、最初のクリーンアップ符号化ステップ308は前処理ステップ412をバイパスし、ステップ420へ直接進む。

【0042】同時に、最初のクリーンアップ符号化ステップ308は判定ブロック414へ進み、そこでストリップ・カウンタ i が2以上でかつ $(M+1)$ 以下であるかどうかのチェックが行われる。但し、 M はコード・ブロック内のストリップの総数である。判定ブロック414がイエス（真）を返した場合、最初のクリーンアップ符号化ステップ308は判定ブロック416へ進み、列カウンタ j が3以上で、かつ $(N+2)$ 以下であるかどうかのチェックが行われる。但し、 N はストリップ内の

列の総数である。判定ブロック416がイエス（真）を返した場合、最初のクリーンアップ符号化ステップ308は、 $(i-1)$ 番目のストリップの $(j-2)$ 番目の列を後処理する後処理ステップ418へ進む。後処理ステップ418は、メモリに既に書き込まれた情報を利用して、係数をMRリストに加えるべきか、それともSPリストに加えるべきかについての決定を行う。この後処理ステップ418について図6を参照しながら以下さらに詳細に後述する。後処理ステップ418の完了後、最初のクリーンアップ符号化ステップ308はステップ420へ進む。判定ブロック414と416のうちのいずれか一方がノー（偽）を返した場合、サブ・プロシージャ308はこの後処理ステップ418をバイパスし、ステップ420へ直接進む。このようにして、後処理ステップ418が $(i-1)$ 番目のストリップの $(j-2)$ 番目の列を処理している間、前処理ステップ412は i 番目のストリップの j 番目の列を処理する。

【0043】ステップ420中、最初のクリーンアップ符号化ステップ308が列カウンタ j を1だけ増分する。次いで、最初のクリーンアップ符号化ステップ308は判定ブロック422へ進み、そこで列カウンタ j が $(N+2)$ 以下であるかどうかのチェックが行われる。但し、 N はストリップ内の列の総数である。判定ブロック422がイエス（真）を返した場合、最初のクリーンアップ符号化ステップ308は次の列 j を処理する判定ブロック408と414へ戻る。一方、判定ブロック422がノー（偽）を返した場合、最初のクリーンアップ符号化ステップ308はステップ424へ進み、そこでストリップ・カウンタ i は1だけ増分される。次いで、最初のクリーンアップ符号化ステップ308が判定ブロック426へ進み、そこでストリップ・カウンタ i が $(M+1)$ 以下であるかどうかのチェックが行われる。但し、 M はコード・ブロック内のストリップの総数である。判定ブロック426がイエス（真）を返した場合、最初のクリーンアップ符号化ステップ308は次のストリップのさらなる処理のためにステップ406へ戻る。

【0044】上述のように、クリーンアップ符号化ステップ308はメモリに書き込まれた情報を読み出して、係数をMRリストに加えるべきか、それとも、SPリストに加えるべきかについての決定を行う。このメモリは、現在のストリップ・レジスタ、前回のストリップ・レジスタ、次のストリップ・ライン・ストア、前回のストリップ・ライン・ストア、および前回のストリップ・ライン・ストアの形になる。

【0045】前回のストリップ・レジスタと現在のストリップ・レジスタとは、4つの行と N 列を有する $4 \times N$ レジスタ配列として各々構成される。この配列は合計で $4N$ のエントリを有する。前のストリップ・レジスタと現在のストリップ・レジスタとからなる本構成はダブル・バッファリング構成として機能する。前のストリップ

・レジスタの4つの行は、それぞれ、コード・ブロックの $(i-1)$ 番目のストリップの連続する4つの行に対応し、前のストリップ・レジスタのN列はコード・ブロックの $(i-1)$ 番目のストリップのN列に対応する。一方、現在のストリップ・レジスタの4つの行はそれぞれ、コード・ブロックの i 番目のストリップの連続する4つの行に対応し、現在のストリップ・レジスタのN列はコード・ブロックの i 番目のストリップのN列に対応する。前のストリップ・レジスタと現在のストリップ・レジスタの各エントリは、コード・ブロックの係数に対応しており、2つのビットN、Sを有する。Sビットは、そのエントリに対応する係数が有意であるかどうかを示す。Nビットはそのエントリに対応する係数が有意係数の 3×3 の近傍に存在するかどうかを示す。現在のストリップ・レジスタは、前処理ステップ412中継続的に更新される。例えば、 i 番目のストリップの j 番目の列の前処理412中、現在のストリップ・レジスタの j 番目の列のSビットが更新され、コード・ブロックの i 番目のストリップの j 番目の列の係数の有効値に応じて対応するNビットが列 $(j-1)$ 、 j 、 $(j+1)$ について更新される。 i 番目のストリップの最後の列の前処理(ステップ424中)の完了後、現在のストリップ・レジスタの内容が前のストリップ・レジスタへ出力され、現在のストリップ・レジスタのN、Sビットはゼロにリセットされる。したがって、 $(i+1)$ 番目のストリップの前処理中の前のストリップ・レジスタには、 i 番目のストリップに対する現在のストリップ・レジスタの完全な更新が含まれる。前処理ステップ412が i 番目のストリップ用の現在のストリップ・レジスタを更新中同時に、後処理418は前のストリップ・レジスタからN、Sビットを読み出している。

【0046】前回のストリップ・ライン・ストアと次のストリップ・ライン・ストアとは、 $1 \times N$ ライン・ストアとして各々構成される。前回のストリップ・ライン・ストアと次のストリップ・ライン・ストアとから成る本構成はダブル・バッファリング構成として機能する。前回のストリップ・ライン・ストアは i 番目のストリップの第1の行に対応する。前回のストリップ・ライン・ストアは、その位置の 3×3 の近傍内の $(i-1)$ 番目のストリップの最後の行に有意係数を持つ位置であって、 i 番目のストリップの第1の行内の非有意係数の位置の列番号を格納する。次のストリップ・ライン・ストアの行は、 $(i+1)$ 番目のストリップの第1の行に対応する。次のストリップ・ライン・ストアは、その位置の 3×3 の近傍内の i 番目のストリップの最後の行に有意係数を持つ位置であって、 $(i+1)$ 番目のストリップの第1の行の非有意係数の位置の列番号を格納する。いずれの場合も、列番号はJPEG2000の走査順でライン・ストアの先頭から次々にライン・ストア内に格納される。いずれの1回についても、次のストリップ・ライ

ン・ストアと前回のストリップ・ライン・ストアの双方ともN列番号まで格納することができる。N未満の列番号しかない場合、ライン・ストアの残りのエントリと最後の部分はブランク状態のままに放置される。次のストリップ・ライン・ストアは、前処理ステップ412中継続的に更新される。例えば、 i 番目のストリップの j 番目の列の前処理412中、 i 番目のストリップの j 番目の列の有意係数の 3×3 の近傍の範囲内に在る、 $(i+1)$ 番目のストリップの第1の行の範囲内のいずれの位置の列番号も、次のストリップ・ライン・ストアに追加される。 i 番目のストリップの最後の列(ステップ424中)の前処理の完了後、次のストリップ・ライン・ストアの内容は前回のストリップ・ライン・ストアへ出力され、次のストリップ・ライン・ストアのエントリはゼロにリセットされる。したがって、 $(i+1)$ 番目のストリップの前処理中の前回のストリップ・ライン・ストアには、 i 番目のストリップに対する次のストリップ・ライン・ストアの完全な更新が含まれる。 i 番目のストリップの処理中の前処理ステップ412は、 $(i+1)$ 番目用のストリップの次のストリップ・ライン・ストアを更新し、 i 番目のストリップの前回のストリップ・ライン・ストアを読み出す。

【0047】前回のストリップ・ライン・ストアは 1×4 ライン・ストアとして構成される。前回のストリップ・ライン・ストアの行は、列 $(j-2)$ 、 $(j-1)$ 、 j 、 $(j+1)$ の $(i-1)$ 番目のストリップの最後の行に対応する。前回のストリップ・ライン・ストアは、その位置の 3×3 の近傍の i 番目のストリップの第1の行に有意係数を持つ位置であって、 $(i-1)$ 番目のストリップの最後の行の非有意係数の位置の列番号を格納する。列番号はJPEG2000の走査順で格納される。前回のストリップ・ライン・ストアは前処理ステップ412中継続的に更新される。例えば、 i 番目のストリップの j 番目の列の前処理412中、 i 番目のストリップの j 番目の列の有意係数の 3×3 の近傍の範囲内にある $(i-1)$ 番目のストリップの最後の行の範囲内の任意の位置の列番号は、(予めそこに存在しない場合)前回のストリップ・ライン・ストアに追加される。同時に、後処理ステップ418は、前回のストリップ・ライン・ストアの第1のエントリを読み出している。これらの列番号は、ライン・ストアの先頭から始まるライン・ストアの前回のストリップ・ライン・ストアの中にJPEG2000の走査順に次々に格納される。いずれの1回についても、前回のストリップ・ライン・ストアは4までの列番号を格納することができる。4未満の列番号しかない場合、前回のストリップ・ライン・ストアの残りのエントリと最後の部分はブランク状態のままに放置される。

【0048】次に図5を参照して、図4に描かれているような i 番目のストリップの j 番目の列の前処理ステッ

ブ412のフローチャートが更に詳細に説明する。前処理ステップ412はステップ502から始まり、ステップ504へ進む。ここで、i番目のストリップのj番目の列の第1のビット・シンボルが検索される。この検索対象ビット・シンボルは、j番目の列でかつi番目のストリップのJPEG2000の走査順で最初（即ち、行1）のビット・シンボルである。次いで、前処理ステップ412はステップ506へ進み、そこでJPEG2000に従って検索対象ビット・シンボルに対する符号化とコンテキスト生成とが行われる。符号化の完了後、ステップ506は必要に応じて有効値テーブルの更新も行う。係数の第1のノンゼロ・ビットプレーン値が符号化されると、係数の有意状態は（1）に変化し、テーブルの配列の対応するエントリは（1）にリセットされる。具体的には、ある係数に属する現在の符号化されたビット・シンボルが（1）である場合、有効値テーブルに格納される係数の有効値は（1）にセットされる。ステップ506の完了後、前処理ステップ412は判定ブロック508へ進み、そこで符号化されたビットが有意な係数に属するかどうかのチェックが行われる。判定ブロック508がイエス（真）を返した場合、前処理ステップ412はステップ510へ進み、そこで検索対象のビット・シンボルに対応するj番目の列であって、現在のストリップ・レジスタのj番目の列のビットSは（1）にセットされる。判定ブロック508がノー（偽）を返した場合（即ち、係数が有意ではない場合）、前処理ステップ412は判定ブロック534へ直接進む。

【0049】ステップ510の後、前処理ステップ412はループ514から526へ進み、そこで現在の有意係数の3×3の近傍内の各非有意近傍が順に処理される。ステップ514中、前処理ステップ412は、（もし存在する場合）は現在の有意係数の非有意近傍を取得する。ステップ514は、有効値テーブルを調べることにより現在の有意係数の非有意近傍を決定する。非有意近傍が存在しなければ、前処理ステップ412は判定ブロック534へ直接進む（図示せず）。一方、非有意近傍が存在する場合、前処理ステップ412は判定ブロック516へ進み、そこで現在の近傍が前回のストリップ（i-1）内にあるかどうかのチェックが行われる。判定ブロック516がイエス（真）を返した場合、前処理ステップ412はステップ518へ進み、そこで現在の近傍の列番号が前回のストリップ・ライン・ストアに追加される。ステップ518の後、前処理ステップ412は判定ブロック526へ進む。一方、判定ブロック516がノー（偽）を返した場合、前処理ステップ412は判定ブロック520へ進み、そこで現在の近傍が次のストリップ（i+1）にあるかどうかのチェックが行われる。判定ブロック520がイエス（真）を返した場合、前処理ステップ412はステップ522へ進み、そこで列番号が次のストリップ・ライン・ストアに追加さ

れる。ステップ522後、前処理ステップ412は判定ブロック526へ進む。一方、判定ブロック520がノー（偽）を返した場合、前処理ステップ412はステップ524へ進む。ステップ524中、前処理ステップ412は、現在の近傍に対応する現在のストリップ・レジスタのエントリ内でビットNを（1）にセットする。ステップ524後、前処理ステップ412は判定ブロック526へ進む。判定ブロック526は、ループ514～526によってまだ処理されていない近傍が、現在の有意係数の3×3の近傍内にさらに存在するか否かをチェックする。判定ブロック526がイエス（真）を返した場合、現在の有意係数の次の近傍の処理を行うために前処理ステップ412はステップ514へ戻る。判定ブロック526がノー（偽）を返した場合、前処理ステップはステップ528へ進む。

【0050】ステップ528中、前処理ステップ412は、JPEG2000の走査順に従う前回のストリップ・ライン・ストアの第1のエントリである前回のストリップ・ライン・ストアの先頭を検索し、それを変数headSPに割り振る。i番目のストリップの最初の前処理ステップ412中（即ち、j=1）、前回のストリップ・ライン・ストアは、i番目のストリップの第1の行に対応し、次いで、それらの3×3の近傍の（i-1）番目のストリップ内の有意係数を持つ位置の列番号を格納する。従って、i番目のストリップの最初の前処理ステップ412（即ち、j=1）中、ある位置のi番目のストリップの第1の行の前回のストリップ・ライン・ストア内の変数headSPにJPEG2000の走査順で格納された第1列番号が含まれる。この位置は（i-1）番目のストリップ内のその位置の3×3の近傍に有意係数を持つ。ステップ528後、前処理ステップ412は判定ブロック530へ進み、そこでj=headSPかどうかのチェックが行われる。判定ブロック530がイエス（真）を返した場合、前処理ステップ412はステップ532へ進み、そこでNビットは位置（1, j）に対応する現在のストリップ・レジスタのエントリ用として（1）にセットされる。このようにして、（i-1）番目のストリップ内の有意係数のi番目のストリップの非有意近傍がそれらの対応するNビットをセットすることにより現在のストリップ・レジスタ内に特定される。また、ステップ532中、前処理ステップ412によって前回のストリップ・ライン・ストアの現在の先頭が取り除かれる。さらに、残りの列番号はすべて前回のストリップ・ライン・ストアの先頭へ一度にシフトされる。ステップ532の完了後、前処理ステップ412は判定ブロック534へ進む。一方、判定ブロック530がノー（偽）を返した場合、前処理ステップ412は判定ブロック534へ直接進む。当業者には明らかなように、前回のストリップ・ライン・ストアはNまでの列番号を最初に格納する。前処理ステップ412が各列j=1, ..., Nについ

て呼び出されると、前処理ステップ412は、判定ブロック530の間、前回のストリップ・ライン・ストア内に対応する列番号が存在するかどうかをチェックする。もし対応する列番号が存在すれば、非有意近傍はその列の第1の行に存在するものとして特定され、現在のストリップ・レジスタに入力され、前回のストリップ・ライン・ストアから取り除かれる。

【0051】判定ブロック534は、前処理ステップ412によってまだ処理されていないビット・シンボルがj番目の列の中に更に存在するかどうかを判定する。判定ブロック534がイエス（真）を返した場合、前処理ステップ412はステップ536へ進み、そこで列内の次のビット・シンボルが検索される。次いで、前処理ステップ412は、この検索されたビット・シンボルを処理するステップ506へ戻る。一方、列jの中にそれ以上処理すべきビット・シンボルが存在しなければ、判定ブロック534はノー（偽）を返し、前処理ステップ412は終了し（538）、さらなる処理を行うために最初のクリーンアップ符号化ステップ308へ戻る。

【0052】次に図6に目を転じると、図4に描かれているような、(i-1)番目のストリップの(j-2)番目の列の後処理ステップ418のフローチャートがさらに詳細に示されている。前述したように、前処理ステップ412がi番目のストリップのj番目の列を処理している間、後処理ステップ418は(i-1)番目のストリップの(j-2)番目の列を処理する。

【0053】後処理ステップ418はステップ602から始まり、ステップ604へ進み、そこで変数position1は(1, j-2)にセットされる。この変数position1は、後処理ステップによって処理される前のストリップ・レジスタ内に現在位置の(行、列)座標を格納する。ステップ604の完了後、後処理ステップ418はステップ610へ進み、そこで前回のストリップ・ライン・ストアの先頭に格納されたエントリが検索され、変数headSPの中に格納される。この変数headSPは、或る位置の3×3の近傍に在るi番目のストリップの中に有意係数を持つ(i-1)番目のストリップの中に位置の列番号を格納する。前回のストリップ・ライン・ストアが空の場合、ステップ606は変数headSPをヌルにセットする。

【0054】ステップ606の完了後、後処理ステップ418は判定ブロック608へ進み、そこで変数headSP=(j-2)かどうかのチェックが行われる。判定ブロック608がイエス（真）を返した場合、後処理ステップ418はステップ610へ進み、そこで変数positon2は(4, headSP)にセットされる。この変数positon2は、(i-1)番目のストリップ内のビット・シンボルの(行、列)を示し、該シンボルはi番目のストリップ内のその3×3の近傍に有意係数を持っている。判定ブロック608がノー（偽）を返した場合、この変数posi

ton2はヌルにセットされる(612)。ステップ612の完了後、後処理ステップ418はステップ614へ進む。

【0055】ステップ614中、後処理ステップ418は変数position1とpositon2を比較し、JPEG2000の走査順でposition1がpositon2より前に存在するかどうかを判定する。存在するならば、後処理ステップ418は変数winning_position内に変数position1を格納する。存在しなければ、後処理ステップ418は変数winning_positionに変数positon2を格納する。ステップ614の完了後、後処理ステップ418は判定ブロック616へ進み、そこで変数winning_positionに格納された座標が変数positon2（即ち、前回のストリップ・ライン・ストア）であるかどうかのチェックが行われる。

【0056】判定ブロック616がイエス（真）を返した場合、後処理ステップはステップ618へ進み、そこでheadSPに対応する前回のストリップ・ライン・ストアの先頭が取り除かれる。また、前回のストリップ・ライン・ストアの残りのエントリは先頭へ1だけ各々シフトされる。ステップ618の完了後、後処理ステップはステップ620へ進み、そこで変数winning_positionに格納された位置に対応する前のストリップ・レジスタ内の位置にあるNビットは(1)にセットされる。ステップ620の完了後、後処理ステップ418は判定ブロック622へ進む。判定ブロック612がノー（偽）を返した場合、後処理ステップは判定ブロック622へ直接進む。

【0057】判定ブロック622は、変数winning_positionに格納された位置に対応する前のストリップ・レジスタ内の位置のSビットが(1)であるかどうかをチェックする。判定ブロック622がイエス（真）を返した場合、後処理ステップ418はステップ624へ進み、そこで変数winning_positionに格納された位置はMRリストに追加される。ステップ624の完了後、後処理ステップ418は判定ブロック630へ進む。判定ブロック622がノー（偽）を返した場合、後処理ステップ418は判定ブロック626へ進む。判定ブロック626は、変数winning_positionに格納された位置に対応する前のストリップ・レジスタ内の位置のNビットが(1)であるかどうかをチェックする。判定ブロック626がイエス（真）を返した場合、後処理ステップ418はステップ628へ進み、そこで変数winning_positionに格納された位置はSPリストに追加される。ステップ628の完了後、後処理ステップ418は判定ブロック630へ進む。判定ブロック626がノー（偽）を返した場合、後処理ステップ418は判定ブロック630へ直接進む。

【0058】判定ブロック630は、(j-2)番目の列の中に処理の対象となる位置がさらに存在するかどうかをチェックする。すなわち、判定ブロック630は、

変数winning_positionに格納された現在位置の行が4に等しいかどうかをチェックする。判定ブロック630がノー（偽）を返した場合、後処理ステップ418はステップ632へ進み、そこで変数position1内の行番号は（1）だけ増分される。ステップ632後、後処理ステップ418はステップ606へ進み、そこで変数position1に格納された新しい値に対して後処理418が再開される。一方、判定ブロック630がイエス（真）を返した場合、後処理ステップ418は終了し（634）、更なる処理を行うためにクリーンアップ符号化ステップ308へ戻る。

【0059】上記から解るように、 i 番目のストリップに対する前処理ステップ412は、現在のストリップ・レジスタ内に格納するために現在の N 、 S ビットを生成する。これらの N 、 S ビットは、 i 番目のストリップ用として前処理ステップの終了時に存在し、次いで、前のストリップ・レジスタへ出力される。 i 番目のストリップの前処理ステップ412と同時に、後処理ステップ418は、前のストリップ・レジスタに現在格納されている（ $i-1$ ）番目のストリップの N 、 S ビットを処理する。この N 、 S ビットは、前回の前処理ステップ412中（ $i-1$ ）番目のストリップ用として生成されたものである。 i 番目のストリップの前処理ステップ412は、 i 番目のストリップ内の有意係数とその非有意近傍とを特定し、現在のストリップ・レジスタの対応する位置の N 、 S ビットとしてそれらを格納する。前処理ステップ412は、（ $i-1$ ）番目のストリップ内の有意係数の i 番目のストリップの第1の行のいずれの非有意近傍も特定し、現在のレジスタの対応する位置に N ビットとしてこれらを格納する。 i 番目のストリップの前処理ステップ412は、前回のストリップ・ライン・ストアに現在格納されている任意のこのような非有意近傍の列番号を検索することにより後者の処理を行う。これらの非有意近傍は（ $i-1$ ）番目のストリップの前回の前処理ステップで特定され、次のストリップ・ライン・ストアに格納され、（ $i-1$ ）番目のストリップの前処理ステップの最後に、前回のストリップ・ライン・ストアへ出力される。 i 番目のストリップの前処理ステップ412によって、 i 番目のストリップの第1の行内の有意係数に起因する非有意近傍であって、（ $i-1$ ）番目のストリップの最後の行内のいずれの非有意近傍も特定される。前処理ステップ412は、これらの非有意近傍の列番号を前回のストリップ・ライン・ストアに追加する。次いで、（ $i-1$ ）番目のストリップに対する後処理ステップ418は、前回のストリップ・ライン・ストアの中からこれらの列番号を検索し、前のストリップ・レジスタ内の任意の対応する N ビットを更新する。（ $i-1$ ）番目のストリップに対する後処理ステップ418は、更新された前のストリップ・レジスタに格納されている N 、 S ビットを利用して、 SP リスト又は MR リス

トに追加が可能な位置を特定する。具体的には、1にセットされた S ビットを有する更新された前のストリップ・レジスタ内の位置は MR リストの中に格納され、 $S=1$ と $N=0$ を持つ更新された前のストリップ・レジスタ内の位置が SP リストの中に格納される。

【0060】図3に戻ると、最初のクリーンアップ符号化ステップ308の完了後、本方法は判定ブロック310へ進み、そこで処理の対象となるビットプレーンがさらに存在するかどうかのチェックが行われる。もし存在すれば、本方法はステップ312へ進み、そこで次のビットプレーンのビット・シンボルが検索される。次いで、本方法は、現在のビットプレーンのシグニフィカンス・プロパゲーション・パスで、関連するビット・シンボルの処理を行うステップ314へ進む。

【0061】次に図7Aと図7Bに目を転じると、図3のシグニフィカンス・プロパゲーション（ SP ）符号化ステップ314のフローチャートがさらに詳細に示されている。この符号化ステップ314は、最初のクリーンアップ符号化ステップ308の後に最初続き、その後、初期ビットプレーンの後の各ビットプレーン用のクリーンアップ符号化ステップ318に続く。 SP 符号化ステップ314は、エントロピ符号化法300と呼ばれサブ・プロセスの形をとる。この符号化ステップ314は、現在の SP リスト（本明細書では、以後オリジナルの SP リストと呼ぶ）が処理を行うために SP 符号化ステップ314へ渡されたところから始まる（702）。前回の最初のクリーンアップ・ステップ308又はクリーンアップ符号化ステップ318によって、このオリジナルの SP リストは生成される。 SP 符号化ステップ314は、 SP 符号化ステップ314へ渡されるオリジナルの SP リストと、近傍 SP リストと、次のストリップ SP リストと、前回のストリップ SP リストと、新しい SP リストの5つのリストを利用する。 SP 符号化パス314によって後者の4つのリストが生成される。近傍 SP リストと、前回のストリップ SP リストと、次のストリップ SP リストとは、それらのリストの利用が SP 符号化ステップ314に限定されるという点でローカルなものである。一方、生成された新しい SP リストと更新されたオリジナルの SP リストとはステップ318中さらなる処理を行うためにエントロピ符号化法へ渡される。

【0062】現在の SP 符号化ステップ214中後で有意になる係数である、このリストに格納されている位置に対応する任意の係数をタグ付けによって特定するために、 SP 符号化ステップ314中オリジナルの SP リストの更新を行うことができる。この目的のために、オリジナルの SP リストの各エントリは、そのロケーション 3×3 の近傍に有意係数を有するロケーションであって、非有意係数の現在のビットプレーン内のビット・シンボルのロケーションを示す（行、列）座標と、現在の

SP符号化ステップ214後にビット・シンボルの有効値を示すフラグとを有する。

【0063】新しいSPリストは、そのロケーションの3×3の近傍に任意の新しい有意係数を持つ非有意係数のビット・シンボルのロケーションの(行、列)座標を有する。これらの新しい有意係数は、現在のビットプレーン内にビット・シンボルを持つ、符号化された有意係数であり、これらの有意係数の有意状態は現在のSP符号化ステップ314中、1に変化する(有意へ変わる)。現在のSP符号化ステップ314中後で有意になる係数である、このリストに格納された位置に対応する任意の係数をタグ付けによって特定するために、SP符号化ステップ314中新しいSPリストの更新を行うことができる。この目的のために、新しいSPリストの各エントリは、そのロケーションの3×3の近傍に新しい有意係数を有するロケーションであって、非有意係数のビット・シンボルのロケーションの(行、列)座標と、現在のSP符号化ステップ314後にビット・シンボルの有効値を示すフラグとを有する。

【0064】近傍SPリストはいくつかのビット・シンボルのコード・ブロックの範囲内にロケーションの(行、列)座標を有する。近傍SPリストは新しいSPリストのサブセットである。すなわち、近傍SPリスト内のすべての位置はこの新しいリストの中に含まれる。近傍SPリストには、SP符号化ステップ314中復号化されたばかりの現在の有意係数の3×3の近傍の範囲内に、非有意係数の位置(但し、必ずしも全ての位置ではない)が含まれる。近傍SPリストは、現在の有意係数と同じコード・ブロックのストリップ内に在り、かつ、JPEG2000の走査順で現在の有意係数の後に在る、3×3の近傍の範囲内の位置しか格納しない。

【0065】次のストリップSPリストは、SP符号化ステップ314中有意であることが判明した、現在のストリップ内の係数の3×3の近傍の範囲内の次のストリップの第1の行の範囲内に非有意係数のロケーションの列座標を有する。次のストリップSPリストは、近傍SPリスト内のすべての位置が新しいリストに含まれるという点で、新しいSPリストのサブセットである。

【0066】前回のストリップSPリストは、SP符号化ステップ314中有意であることが判明した、前回のストリップ内の係数の3×3の近傍の範囲内の現在のストリップの第1の行の範囲内に非有意係数のロケーションの列座標を有する。

【0067】開始後、SP符号化ステップ314は初期化ステップ704へ進み、そこで近傍SPリストに対するエントリと、次のストリップSPリストと、前回のストリップSPリストと、新しいSPリストとがゼロ

(0)にリセットされる。初期化ステップ704後、サブ・プロシージャは判定ブロック706へ進み、そこでオリジナルのSPリスト内に何らかの位置が存在するか

どうかのチェックが行われる。判定ブロック706がノー(偽)を返した場合、SP符号化ステップ314は終了し(714)、エントロピ符号化法はMR符号化ステップ316へ進む。コード・ブロックの範囲内のすべての位置が有意で、MR符号化ステップ316中符号化の対象となる場合、上記の事態が生じることもある。判定ブロック706がイエス(真)を返した場合、SP符号化ステップ314はループ(ステップ710~728)と、クリーンアップ・リスト生成ステップ708とへ同時に進む。SP符号化ステップ314がSP位置を符号化し、上述のSPリストを生成している間、クリーンアップ・リスト生成ステップ708はクリーンアップ・リストを生成する。以下より詳細にクリーンアップ・リスト生成ステップ708について説明する。

【0068】ループ(710~728)は、ループの各パスについてオリジナルのSPリスト内と、近傍SPリスト内と、前回のストリップSPリスト内の各位置を処理する。ループの任意のパス中にSP符号化ステップ314は、オリジナルのSPリスト内と、近傍SPリスト内と、前回のストリップSPリスト内の次の位置であって、前回のパス中検索されなかった次の位置を検索する(720)。これらのリスト内の位置は、JPEG2000の走査順に従ってループのパス中次々に検索される。これらの位置の検索方法について図8を参照しながら以下さらに詳細に後述する。一方、これらのリストの範囲内に前に検索されなかった位置がそれ以上存在しなかった場合、SP符号化ステップ314はループ(ステップ710~728)から抜けてステップ712へ進む。

【0069】初期化ステップ704の完了後、SP符号化ステップ314は判定ブロック706へ進み、そこでオリジナルのSPリスト内、近傍SPリスト内及び前回のストリップSPリスト内に位置がさらに存在するかどうかのチェックが行われる。これらのリストの範囲内に残りの位置が存在すれば、判定ブロックはイエス(真)を返し、SP符号化ステップ314は判定ブロック710へ進む。

【0070】判定ブロック716はコード・ブロックの現在のストリップ内、オリジナルのSPリスト内、近傍SPリスト内及び前回のストリップSPリスト内に、前に検索されなかった位置がさらに存在するかどうかをチェックする。SP符号化ステップ314は、第1のストリップからコード・ブロックのストリップを処理し、JPEG2000に記載されている走査順で、最後のストリップまでずっと処理を継続する。判定ブロック716の最初のパス中、現在のストリップはコード・ブロックの第1のストリップである。判定ブロック716がノー(偽)を返した場合、SP符号化ステップ314はステップ718へ進み、そこで次のストリップSPリスト内の列位置が前回のストリップSPリストへ出力され、次

のストリップSPリストのエントリはゼロにセットされる。前述したように、次のストリップSPリストは、SP符号化ステップ314中有意であることが判明した係数である、現在のストリップ内の係数の3×3の近傍の範囲内の次のストリップの第1の行の範囲内の非有意係数のロケーションの列座標を有する。前回のストリップSPリストは、SP符号化ステップ314中有意であることが判明した係数である、前回のストリップ内の係数の3×3の近傍の範囲内の現在のストリップの第1の行の範囲内の非有意係数のロケーションの列座標を有する。SP符号化ステップ314は、次のストリップSPリストに列番号を追加し（ステップ738）、SP符号化コード・ブロックの任意のストリップの処理中、前回のストリップSPリストの中から列番号を検索する。現在のストリップの範囲内にそれ以上位置が存在しなければ、次のストリップSPリストの列番号が前回のストリップSPリストへ出力され（718）、次のストリップSPリストのエントリがゼロにセットされ、次のストリップが現在のストリップになる。ステップ718の完了後、SP符号化ステップ314はステップ720へ進む。

【0071】一方、判定ブロック716がイエス（真）を返した場合、SP符号化ステップ314はステップ720へ直接進む。ステップ720中、SP符号化ステップ314はオリジナルのSPリスト内と、近傍SPリスト内と、前回のストリップSPリスト内の次の位置であって、ループの前回のパス中検索されなかった位置を検索する。以下これについて図8を参照しながらより詳細に説明する。ステップ720の完了後、SP符号化ステップ314はステップ722へ進む。そこでコンテキスト生成とビット・シンボルの符号化とがその（検索）位置において行われる。この符号化とコンテキスト生成とは、JPEG2000に従って好適に行われる。符号化の完了後、ステップ722は必要な場合に有効値テーブルの更新も行う。係数の第1のノンゼロ・ビットプレーン値が符号化され、テーブルの配列の対応するエントリが（1）にリセットされた場合、係数の有意状態は（1）に変化する。具体的には、ある係数に属する現在の符号化されたビット・シンボルが（1）である場合、有効値テーブルに格納されるその係数の有効値は（1）にセットされる。

【0072】符号化ステップ722の完了後、SP符号化ステップ314は判定ブロック724へ進む。判定ブロック724では、その（検索）位置における符号化されたビット・シンボルが（1）の有意状態を持っているかどうかのチェックが行われる。判定ブロック724がノー（偽）を返した場合、SP符号化ステップ314はステップ710へ戻り、そこで、オリジナルのSPリスト内と、近傍SPリスト内と、前回のストリップSPリスト内の次の位置であって、ループの前回のパス中検索

されなかった次の位置が検索される。一方、判定ブロック724がイエス（真）を返した場合、SP符号化ステップ314はステップ726へ進む、そこでオリジナルのリスト内又は新しいSPリスト内の対応する（検索された）位置のフラグに有効なものとしてタグが付けられる。このようにして、次のクリーンアップ符号化ステップ318中、MRリストへの後続する追加を行うためにこの位置が特定される。

【0073】ステップ726の完了後、SP符号化ステップ314は、新しいSPに非有意近傍を追加するための処理へこの非有意近傍の（行、列）座標を渡す（732）。これについては図13Aと図13Bを参照しながらさらに詳細に後述する。この処理はSP符号化ステップ314の開始から始まり、SP符号化ステップ314の完了で終了する。

【0074】ステップ726の完了後、SP符号化ステップ314はサブループ728～738、740へ進む、そこで現在の符号化されたビット・シンボルを有する現在の有意係数の3×3の近傍内の非有意係数がサブループの各パスについて順に処理される。サブループ728～738、740は判定ブロック728を有し、この判定ブロックで、サブループによって前に処理されなかったような、現在の有意係数の非有意近傍がさらに存在するかどうかのチェックが行われる。判定ブロック728は有効値テーブルを参照してこの判定を行う。判定ブロック728がノー（偽）を返した場合、SP符号化ステップ314はさらなる処理を行うために判定ブロック710へ戻る。一方、判定ブロック728がイエス（真）を返した場合、SP符号化ステップ314はステップ730へ進む、そこで（行、列）座標としての非有意近傍の位置が有効値テーブルから検索される。ステップ730の完了後、SP符号化ステップ314は判定ブロック734へ進む、そこで現在の非有意近傍の位置が、JPEG2000の走査順で現在の符号化されたビット・シンボルを有する係数の位置の後に在るかどうかのチェックが行われる。判定ブロック734がノー

（偽）を返した場合、さらなる処理を行うために、SP符号化ステップ314は判定ブロック728へ戻る。一方、判定ブロック734がイエス（真）を返した場合、SP符号化ステップ314は判定ブロック736へ進む、そこで現在の符号化されたビット・シンボルが属する相手先の現在のストリップ後、現在の非有意近傍が次のストリップに属するかどうかのチェックが行われる。判定ブロック736がイエス（真）を返した場合、SP符号化ステップ314はステップ738へ進む、そこで現在の非有意近傍の位置の列番号が次のストリップSPリストの最後に追加される。ステップ738の完了後、SP符号化ステップ314はさらなる処理を行うために判定ブロック728へ戻る。一方、判定ブロック738がノー（偽）を返した場合、SP符号化ステップ314

はステップ740へ進み、そこで現在の非有意近傍の位置の(行、列)座標が近傍SPリストの最後に追加される。ステップ740の完了後、SP符号化ステップ314はさらなる処理を行うために判定ブロック728へ戻る。

【0075】次に図8に目を転じると、SP符号化ステップ314のステップ720のフローチャートがさらに詳細に示されている。ステップ720はJPEG2000の走査順に従って、オリジナルのSPリスト内、近傍SPリスト内、または、前回のストリップSPリスト内に格納される第1の位置であって、前に検索され、処理されなかった第1の位置を検索する。以上から明かなように、オリジナルのSPリスト内、近傍SPリスト内、及び、前回のストリップSPリスト内に格納された位置は、JPEG2000に従ってソートされる。ステップ720に対する第1のコール中、3つのヘッド・ポインタはそれぞれ、オリジナルのSPリストへの第1のエントリと、近傍SPリストへの第1のエントリと、前回のストリップSPリストへの第1のエントリを指している。各後続するコールについて、これらのヘッド・ポインタの中の1つはリスト内の次の項目へ移動する。ステップ718が次のストリップSPリストの内容を前回のストリップSPリストへ出力した場合、前回のストリップSPリストのためのヘッド・ポインタは、前回のストリップSPリストの第1のエントリにリセットされる。ヘッド・ポインタの位置は、現在のエントロピ・ステップ314内の、ステップ720への各コールの間で保持される。現在のエントロピ・ステップ314が完了されるとすぐにヘッド・ポインタはリセットされる。

【0076】ステップ720はステップ802から始まり、判定ブロック804、806、808へ同時に進む。判定ブロック804で、ステップ720は、ヘッド・ポインタがオリジナルのSPリスト内の有意な(即ち、ヌルでない)位置を指しているかどうかをチェックする。判定ブロック804がイエス(真)を返した場合、ステップ720はステップ812へ進み、そこでステップ720は、ヘッド・ポインタが指しているオリジナルのSPリスト内のその位置の(行、列)座標を検索し、変数origSPにこの座標を格納する。一方、判定ブロック804がノー(偽)を返した場合、ステップ720は変数origSPをヌルにセットする。判定ブロック806で、ステップ720はヘッド・ポインタが近傍SPリスト内で有意な(即ち、ヌルでない)位置を指しているかどうかをチェックする。判定ブロック806がイエス(真)を返した場合、ステップ720はステップ816へ進み、そこでステップ720は、ヘッド・ポインタが指している、近傍SPリスト内のその位置の(行、列)座標を検索し、変数neighSPにこの座標を格納する。一方、判定ブロック806がノー(偽)を返した場合、ステップ720は変数neighSPをヌルにセットする。判定

ブロック808で、ステップ720はヘッド・ポインタが前回のストリップSPリストの中で有意な(即ち、ヌルでない)位置を指しているかどうかをチェックする。判定ブロック808がイエス(真)を返した場合、ステップ720はステップ820へ進み、そこでステップ720はヘッド・ポインタが指している前回のストリップSPリスト内のその位置の列番号を検索する。この列番号は変数nextstripSP内の座標(1、列番号)として格納される。上述のように、前回のSPリストは、前回のストリップのSP符号化ステップ314中符号化された有意係数に起因する非有意近傍であって、ストリップの第1の行の任意の非有意近傍の列番号を格納する。一方、判定ブロック808がノー(偽)を返した場合、ステップ720は変数nextstripSPをヌルにセットする。

【0077】ステップ810~820の完了後、ステップ720はステップ822へ進む。ステップ822中、変数origSPと、neighSPと、nextstripSPとに格納された位置のロケーションが比較され、これらの3つの位置から成る位置であって、JPEG2000の走査順で第1となる当該位置が発見される。ステップ822の完了後、ステップ720はステップ824へ進み、そこでこの‘第1の’位置を含むエントリへのヘッド・ポインタは、対応するリスト内でこの第1のエントリから離れた方向へ1エントリだけ移される。比較ステップ822によって得られるこの‘第1の’位置は、次いで、さらなる処理を行うために、SP符号化ステップ314の符号化ステップ722とクリーンアップ・リスト生成ステップ708とへ渡される。次いで、ステップ720は終了する。

【0078】SP符号化ステップ314がステップ710~740を実行中している間、SP符号化ステップ314はクリーンアップ・リスト生成ステップ708を実行する。これについて以下より詳細に説明する。

【0079】次に図13Aと図13Bに目を転じると、非有意近傍を新しいSPリストに追加する処理を示すフローチャートが示されている。SP符号化ステップ314は、SP符号化パス314中有意であることが判明した係数の、非有意近傍の(行、列)座標をステップ732(図7A)中に渡す。これらの位置は、非有意近傍を新しいSPリストに追加するための処理1300へ渡される。この処理1300は、SP符号化ステップ314の開始から始まり、SP符号化ステップ314の完了で終了する。この処理1300の目的は、有意係数の非有意近傍の任意のオーバーラップ部分を取り除くことである。例えば、2つの有意係数からなる非有意近傍である係数が存在するため、新しいSPリストへの追加を1回しか必要としない場合がある。これらの非有意近傍は、新しいSPリストに追加され、JPEG2000の走査順にSPリスト内でソーティングが行われる。処理1300は、4×Nの前のストリップ・レジスタと、4×N

の現在のストリップ・レジスタと、 $1 \times N$ の前のストリップ・ライン・ストアと、 $1 \times N$ の次のストリップ・ライン・ストアと、前述した 1×4 の前のストリップ・ライン・ストアとを利用する。但し、 N はブロックの幅である。処理1300は、最初のクリーンアップ符号化パス308で用いられたような前処理ステップに多少類似した方法で前処理ステップとして機能する。処理1300は、前のストリップ・レジスタと前のストリップ・ライン・ストアとの内容を後処理プロセス1400へ送る。これについては図14を参照してさらに詳細に後述する。この後処理プロセス1400は、最初のクリーンアップ符号化パス308で用いられた方法と多少類似した方法で後処理ステップとして機能する。例えば、処理1300が i 番目のストリップを前処理している間、処理1400は $(i-1)$ 番目のストリップを後処理していることになる。

【0080】前処理プロセス1300は、 i 番目のストリップ内の有意係数の 3×3 の近傍の範囲内に存在する非有意係数である、同じ i 番目のストリップ内の対応する非有意係数用の $4 \times N$ の現在のストリップ・レジスタのエントリの N ビットを(1)にセットする。前処理プロセス1300はまた、 $(i-1)$ 番目のストリップの最後の行に在り、かつ、 i 番目のストリップの第1の行の有意係数の 3×3 の近傍に在る非有意係数の列番号を前のストリップ・ライン・ストア内に格納する。一旦、 i 番目のストリップが前処理されると、該 i 番目のストリップは後処理を行うために処理1400へ送られる。処理1400はまた前のストリップ・ライン・ストア内のエントリであって、 $(i+1)$ 番目のストリップの前処理中、前のストリップ・ライン・ストアに現在追加されているエントリをフェッチする。即ち、後処理するプロセス1400は、 i 番目のストリップの最後の行内に在り、かつ $(i+1)$ 番目のストリップの第1の行の有意係数内の 3×3 の近傍に在る非有意係数の列番号を順にフェッチする。

【0081】処理1300はステップ1302から始まり、そこで、上述のレジスタのエントリとライン・ストアとはゼロにセットされる。次いで、処理1300はステップ1304へ進み、そこでステップ732中に渡された有意係数の(行、列)座標がフェッチされる。このフェッチされた(行、列)座標は変数CurPosに格納される。次いで、処理1300は判定ブロック1306へ進み、そこでCurPosに格納された、位置の 3×3 の近傍内に非有意係数が存在するかどうかのチェックが行われる。

【0082】判定ブロック1306がイエス(真)を返した場合、処理1300はループ1312~1324へ進み、そこで位置CurPosの 3×3 の近傍内の非有意近傍が順にフェッチされ、処理される。具体的には、処理1300はステップ1312へ進み、そこで次の非有意近

傍の(行、列)座標がフェッチされ、変数NextNに格納される。次いで、処理1300は判定ブロック1314へ進み、そこで位置NextNが現在のストリップの前のストリップの中に存在するかどうかのチェックが行われる。判定ブロック1314がイエス(真)を返した場合、位置NextNの列番号が、前のストリップ・ライン・ストアに追加される(1316)。イエス(真)を返さなかった場合、処理1300は判定ブロック1320へ進み、そこで位置NextNが、現在のストリップの次のストリップの中に存在するかどうかのチェックが行われる。判定ブロック1320がイエス(真)を返した場合、位置NextNの列番号は次のストリップ・ライン・ストアに追加される(1318)。一方、判定ブロック1320がノー(偽)を返した場合、処理1300は、位置NextNに対応する現在のレジスタ内のエントリに対してビットNを(1)にセットする。ステップ1316、1318、又は1322の完了後、処理1300は判定ブロック1324へ進む。判定ブロック1324で、非有意近傍がさらに存在するかどうかのチェックが行われる。判定ブロック1324がイエス(真)を返した場合、処理1300はステップ1312へ戻り、次の非有意近傍のさらなる処理を行う。イエス(真)を返さなかった場合ステップは1328へ続く。

【0083】ステップ1328で、変数PrevPosは変数CurPosにセットされる。次いで、処理1300はステップ1330へ続き、そこで、処理1300はステップ732(図7A)中に渡された次の位置の(行、列)座標をフェッチする。このフェッチされた(行、列)座標は変数CurPosに格納される。この次の位置がまだ利用可能でない場合、処理1300は利用可能になるまで待機する。それ以上利用可能な位置が存在しなければ、処理1300はステップ1334へ直接進む(図示せず)。

【0084】ステップ1330の完了後、処理は判定ブロック1332へ進み、そこで変数CurPos内に格納された位置が、変数PrevPosに格納された位置に関して、次のストリップ内に存在するかどうかのチェックが行われる。判定ブロック1332がノー(偽)を返した場合、処理1300は判定ブロック1306へ戻り、変数CurPosの中に格納された新しい位置のさらなる処理が行われる。判定ブロック1332がイエス(真)を返した場合、処理1300はステップ1334へ続く。

【0085】ステップ1334中、処理1300は現在のストリップ・レジスタの内容を前のストリップ・レジスタへ移動させ、現在のストリップ・レジスタ内のエントリをゼロにセットする。ステップ1334は、前のストリップ・ライン・ストアへ次のストリップ・ライン・ストアの内容を移動させ、次いで、次のストリップ・ライン・ストア内のエントリをゼロにセットする。ステップ1334の完了後、処理はステップ1336と1338とへ同時に続く。ステップ1336中、処理1300

0は、前のストリップ・レジスタを後処理する。これについては以下図14を参照しながらさらに詳細に説明する。ステップ1338中、処理1300は、前回のストリップ・ライン・ストア内に現在格納されている位置に対応する現在のストリップ・レジスタ内のエン트리用としてビットNを(1)にセットする。ステップ1336と1338との完了後、処理は、非有意近傍をさらに処理するために判定ブロック1306へ続く。

【0086】判定ブロック1306がノー(偽)を返した場合、即ち、非有意近傍が存在しない場合、処理1300は判定ブロック1308へ続く。判定ブロック1308は、ステップ732(図7A)によって渡された位置がさらに存在するかどうかのチェックを行う。判定ブロック1308がイエス(真)を返した場合、処理1300はステップ1304へ進み、そこでその位置はフェッチされる。現在利用可能な位置が存在しない場合には、判定ブロック1308は、位置が生じるまで待機するか、あるいは、SP符号化パス314は、処理が終了する(1310)終了時刻にSP符号化パス314が終了するまで待機する。

【0087】次に図14に目を転じると、図13Aと13Bの前処理1300と共に使用される後処理1400のフローチャートが示されている。後処理ステップ1400はSP符号化パス314と同時に始まる。後処理1400は1402を開始し、ステップ1336中に後処理1400へ渡される前のストリップ・レジスタを待機する(1404)。前のストリップ・レジスタが後処理1400へ渡された後、処理1400はループ1406~1416へ進む。ループ1406~1416は、前のストリップ・レジスタ内の各エン트리(即ち、位置)であって、(1)にセットされたNビットを持つ各エントリを順に処理し、前回のライン・ストア内に現在格納されているエン트리(列番号)とそのエントリとを比較し、JPEG2000の走査順での以前の位置を新しいリストSPに追加する。ループ1406は、JPEG2000の走査順に前のストリップ・レジスタ内のエントリを処理する。上記から明らかなように、前処理1300は、エントリを追加することにより後処理1400中も前回のライン・ストアを継続的に更新している。

【0088】ループ1406~1416は判定ブロック1406とステップ1408とを有する。判定ブロック1406で、処理1400は、(1)にセットされたNビットを持つ、前のストリップ・レジスタ内に何らかのエン트리(即ち、位置)が、または、前回のストリップ・ライン・ストア内に何らかの位置が存在するかどうかのチェックを行う。判定ブロック1406がイエス(真)を返した場合、処理1400はステップ1408と1410へ同時に進む。ステップ1408で、処理1400は、現在前回のライン・ストア内に在る次の(第1の)列番号をフェッチし、変数LinePosに該列番号を

格納をする。前回のライン・ストア内に現在エントリが存在しなければ、変数LinePosはヌルにセットされる。同時に、ステップ1410は、前にフェッチされなかった次の位置であって、(1)にセットされたNビットを持つ次の位置を前のストリップ・レジスタからJPEG2000の走査順にフェッチし、変数StripPosにその位置を格納する。ステップ1408と1410の後、処理1400はステップ1412へ進み、そこで処理は、変数LinePosとStripPos内に格納された位置の比較を行う。ステップ1412は、変数LinePosとStripPosに格納された該位置であって、JPEG2000の走査順で早い位置にある該位置を決定し、変数WinPosにその位置を格納する。ステップ1412の完了後、処理はステップ1414へ進み、そこでその位置が前回のライン・ストア内に存在する場合、変数WinPosに格納された位置は、前回のライン・ストアから取り除かれる。ステップ1416の完了後、処理はステップ1416へ進み、そこで変数WinPosに格納された位置は新しいSPリストに追加される。処理1400は更なる処理を行うために判定ブロック1406へ戻る。判定ブロック1406がノー(偽)を返した場合、処理1400は終了する(1418)。

【0089】次に図9に目を転じると、SP符号化ステップ314のクリーンアップ・リスト生成ステップ708のフローチャートが示されている。クリーンアップ・リスト生成ステップ708はステップ902から始まり、ステップ904へ進み、そこでループの第1のパス(710~740)中ステップ720で定められた位置を取得する。この位置がJPEG2000の走査順に従う第1の位置であり、この位置は、オリジナルのSPリスト、または近傍SPリスト、または前回のストリップSPリストのいずれかの中に格納される。この位置は変数headSP内に格納される。また、ステップ904は、MRリストに格納されている第1の位置をJPEG2000の走査順に従って取得し、変数headMRの中にこの位置を格納する。ステップ904の完了後、クリーンアップ生成ステップ708はループ906へ進む。

【0090】ループ906は、JPEG2000の走査順でコード・ブロックの各位置 $P(x, y)$ を通して1だけ増分する。コード・ブロックの新しい位置 $P(x, y)$ に対する各々の増分後、ループ906は判定ブロック908へ進む。判定ブロック908は、ループの現在位置 $P(x, y)$ が変数headSP内に格納された位置に等しいかどうかをチェックする。判定ブロックがノー(偽)を返した場合、クリーンアップ生成ステップ708は判定ブロック912へ進み、そこでループの現在位置 $P(x, y)$ が、変数headMRに格納された位置に等しいかどうかのチェックが行われる。判定ブロックがノー(偽)を返した場合、クリーンアップ生成ステップ708はステップ916へ進み、そこで現在位置 $P(x, y)$ がクリーンアップ・

リストに追加される。ステップ916の完了後、クリーンアップ生成ステップ708はループ906へ戻り、そこで、位置P(x,y)はJPEG2000の走査順で次の位置へ増分される。

【0091】一方、判定ブロック908がイエス（真）を返した場合、クリーンアップ生成ステップ708はステップ910へ進む。ステップ910中、ステップ720で定められた次の位置が利用可能な場合、クリーンアップ生成ステップ708はその位置を取得し、変数headSPにその位置を格納する。このようにして、クリーンアップ生成ステップ708は、ステップ720のパス中に定められた位置を次々に検索し、ステップ910をコールする度に、変数headSPに格納された現在位置を次の位置と置き換える。次の位置が現在利用できない場合が時としてある。そのような状況では、クリーンアップ生成ステップ708はその位置が利用可能になるまで待機する。さらに、判定ブロック710（図7A）がノー（偽）を返した場合、クリーンアップ生成ステップ708は、待機する代わりに、このステップ910中に、変数headSPをヌルにセットする。後者の場合、SP符号化ステップ中符号化の対象となる位置はそれ以上存在しない。しかし、SP符号化ステップ中、クリーンアップ・リストに追加する必要がある、符号化の対象となる位置が最後の位置の後に存在する可能性がある。変数headSPをヌルにセットすることにより、ループ906の後続パス中、これらの位置をクリーンアップ・リストに追加することができる。ステップ910の完了後、クリーンアップ生成ステップはループ906へ戻り、そこで、位置P(x,y)はJPEG2000の走査順で1だけ増分される。

【0092】さらに、判定ブロック912がイエス（真）を返した場合、クリーンアップ生成ステップ708はステップ914へ進む、そこでMRリスト内の次の位置が変数headMRに格納される。このようにして、クリーンアップ生成ステップはMRリスト内の位置を次々に検索し、変数headMRに格納された現在位置をステップ914の各コールを用いて次の位置と置き換える。MRリストに残された位置がそれ以上存在しなければ、ステップ912は変数headMRをヌルにセットする。ステップ914の完了後、クリーンアップ生成ステップはループ906へ戻り、位置P(x,y)は、JPEG2000の走査順で1だけ増分される。

【0093】コード・ブロックの範囲内のすべての位置P(x,y)がクリーンアップ生成ステップ708によって処理された後クリーンアップ生成ステップ708は終了する。

【0094】このようにして、現在位置が現在のSPパス内で処理されている間、その位置はクリーンアップ・リスト生成ステップ708へも送られる。このステップ708では、JPEG2000の走査順にブロック内の

すべての座標を通るループが存在する。生成された座標は現在のSP符号化パス内の次の位置と比較される。上記座標と位置とが等しければ、その位置は現在のパス内で予め処理され、その処理が続けられて現在のSP符号化パス内に次の位置が取得され、その処理は次の座標を生成する。現在のSP符号化パスから得られる次の位置が存在しなければ、位置が生じるまで、或いは、SP符号化が完了するまで処理は待機する。現在の座標が変数headSPに等しくない場合、現在の座標は変数headSPに格納されている座標より前の位置に存在し、したがってこの座標はクリーンアップ・リストの候補である。次いで、この座標は変数headMRと比較される。上記座標と位置とが等しくない場合、その座標は既にMRリストに在るため、処理はMRリスト内の次の位置を取得し、変数headMR内にそれを格納し、次の座標を生成する。この生成された座標が変数headMRに等しくない場合、この座標はクリーンアップ・リストの中へ挿入される。一旦、全ての位置が符号化されてしまうと、変数headSPをヌルにセットすることができる。同様に、いったんMRリストの全てのMR位置が処理されてしまった場合、変数headMRはヌルにセットすることができる。このようにして、SPリストとMRリストの最後の後の座標はクリーンアップ・リストに追加することができる。

【0095】本構成の変更例では、処理が、ストリップ内の或る列の4つの位置がすべてクリーンアップ・リストに在ることを検出した場合、処理はクリーンアップ・リストの中で1つの合成座標にその4つの位置を圧縮することができる。この合成座標は、符号化が可能な場合、後の符号化／復号化のクリーンアップ段階で利用することができる。

【0096】図7Aに戻って、終了ステップ714中、SP符号化ステップ314はクリーンアップ符号化パス318中に新しいSPリストと更新されたオリジナルのSPリストとを返す。一方、エントロピ符号化法300はマグニチュード・リファインメント符号化パス316へ進む。

【0097】次に図10に目を転じると、マグニチュード・リファインメント符号化パス316のフローチャートがさらに詳細に示されている。マグニチュード・リファインメント符号化パス316はステップ1002から始まり、ループ1004～1008へ進む、そこでループ1004～1008のそれぞれのパス用のMRリスト内の位置がJPEG2000の走査順で検索される。ステップ1002後、マグニチュード・リファインメント符号化パス316は、MRリスト内の第1の位置の処理を行うためにループのステップ1004と1006とへまず進む。ステップ1006の完了後、マグニチュード・リファインメント符号化パス316は判定ブロック1008へ進む、そこで前に処理されなかった、処理の対象となる位置がMRリストの中にさらに存在するかどうか

かのチェックが行われる。判定ブロック1008がイエス（真）を返した場合、マグニチュード・リファインメント符号化パス316はステップ1004へ戻り、そこで、MRリストの次の位置が検索される。ステップ1004中、マグニチュード・リファインメント符号化パス316は、MRリストから現在位置をフェッチし、この現在位置を用いてメモリから対応するシンボルをフェッチする。マグニチュード・リファインメント符号化パス316はまた、現在位置の3×3の近傍内の有意値を有効値テーブルからフェッチする。ステップ1004の完了後、マグニチュード・リファインメント符号化パス316はステップ1006へ進む。ステップ1006中、これらの有効値はコンテキスト生成処理へ送出されてコンテキストが生成され、対応するシンボルの符号化が行われ、さらに、そのシンボルに対して算術符号化が行われる。これらのステップ1004と1006とは1サイクル当たり1つの位置のレートで行うことができる。符号化処理では、ビットプレーン・メモリから必要なデータ・ビットをフェッチするために位置の座標が利用される。復号化処理では、出力用バッファの中に復号ビットを書き込むために位置の座標が利用される。判定ブロック1008がノー（偽）を返した場合、マグニチュード・リファインメント符号化パス316は終了し（1010）、次いで、エントロピ符号化法はクリーンアップ符号化パス318へ進む。

【0098】次に図11に目を転じると、クリーンアップ符号化パス318のフローチャートが示されている。クリーンアップ符号化パス318はステップ1102から始まる。このステップ1102中、前回のシグニフィカンス・プロパゲーション符号化パス314中生成されたクリーンアップ・リストは現在のクリーンアップ符号化パス318へ渡される。また、前回のクリーンアップ符号化パス308または318中に生成され、前回の符号化パス314中に更新されたオリジナルのSPリストは、クリーンアップ符号化パス318へ渡される。さらに、前回の符号化パス314中生成された新しいSPリストは、クリーンアップ符号化パス318へ渡される。開始ステップ1102中、クリーンアップ符号化パス318は、位置の（行、列）座標を格納するためのクリーンアップ有意係数リストとクリーンアップ非有意近傍リストと呼ばれる2つのリストも初期化し、それらのリストのエントリをヌルにセットする。最後に、前回のクリーンアップ符号化パス308または318中に生成されたMRリストは現在のクリーンアップ符号化パス318へ渡される。

【0099】開始ステップ1102後、クリーンアップ符号化パス318はループ1104～1114へ進み、そこでクリーンアップ・リスト内の位置は、JPEG2000の走査順でループ1104～1114のそれぞれのパス中検索される。ステップ1102後、クリーンア

ップ符号化パス318は判定ブロック1104へ進み、そこで現在のクリーンアップ符号化パス318によって検索されなかった位置がクリーンアップ・リストの中にさらに存在するかどうかのチェックが行われる。判定ブロック1104がイエス（真）を返した場合、クリーンアップ符号化パス318はステップ1106へ進み、そこで前に検索されなかった、クリーンアップ・リスト内の次の位置がフェッチされる。これらの位置は、JPEG2000の走査順でクリーンアップ・リストから検索される。この位置がフェッチされた後、クリーンアップ符号化パス318はステップ1108へ進む。ステップ1108中、クリーンアップ符号化パス318はこの現在位置を利用して、その位置のビットプレーン・メモリから対応するビット・シンボルをフェッチする。クリーンアップ符号化パス318は、現在位置の3×3の近傍の有効値テーブルから有効値もフェッチする。さらに、これらの有意値はコンテキスト生成処理へ送出され、現在のシンボルの符号化用コンテキストが生成される。次いで、この現在のシンボルは算術符号化される。

【0100】ステップ1108の完了後、クリーンアップ符号化パス318は判定ブロック1110へ進み、そこで符号化された現在のビット・シンボルが有意であるかどうかのチェックが行われる。判定ブロック1110がノー（偽）を返した場合、クリーンアップ符号化パス318は、クリーンアップ・リスト内の次の位置（もし存在する場合）の処理を行うために判定ブロック1104へ戻る。一方、判定ブロック1110がイエス（真）を返した場合、クリーンアップ符号化パスはステップ1112へ進み、そこで現在位置の（行、列）座標がクリーンアップ有意係数リストに追加される。クリーンアップ有意係数リスト内の位置はJPEG2000の走査順に格納される。ステップ1112の完了後、クリーンアップ符号化パス318はステップ1114へ進む。

【0101】ステップ1114中、クリーンアップ符号化パス318は、現在位置の3×3の近傍の範囲内の非有意係数を決定し、これらの非有意近傍の位置をクリーンアップ非有意近傍リストに追加する。クリーンアップ非有意近傍リスト内の後者の位置はJPEG2000の走査順に従ってソートされる。当業者には明らかなように、クリーンアップ非有意近傍リストは、クリーンアップ・リスト自身にない係数の位置を有することができる。これらの非有意係数をクリーンアップ非有意近傍リストに追加する方法は、新しいSPリストへの非有意係数の追加と関連して説明した方法（図13A、13B、14）と類似している。前者の場合、クリーンアップ符号化パス318は、図13A、13B、14を参照しながら説明したのと同じ処理を利用するが、以下の変更を伴う。クリーンアップ符号化パス318によって現在符号化されている有意係数の（行、列）座標は、ステップ1114中ステップ1304（図13A）へ渡され、

代わりにステップ1414で変数WinPosがクリーンアップ非有意近傍リストに追加される。このようにして非有意係数のいずれのオーバーラップも取り除かれる。

【0102】ステップ1114の完了後、クリーンアップ符号化パス318はJPEG2000に従って符号ビットを符号化し（図示せず）、次いで、判定ブロック1104へ戻る。判定ブロック1104がノー（偽）を返した場合、即ち、検索され、処理されたクリーンアップ・リストに位置がそれ以上存在しない場合、クリーンアップ符号化パス318はステップ1116へ進み、そこでクリーンアップ符号化パス318は終了する。

【0103】クリーンアップ符号化パス318がクリーンアップ・リスト内の各位置の処理を行っている間と同時に、クリーンアップ符号化パスは、次のSP符号化パス314とMR符号化パス316中、後続する処理を行うために、新しいオリジナルのSPリストと新しいMRリストとを生成する（1116）。

【0104】次に図12に目を転じると、新しいオリジナルのSPリストと新しいMRリストとを生成するステップ1116のフローチャートがさらに詳細に示されている。ステップ1116は1202から始まり、そこで、更新されたオリジナルのSPリストと、新しいSPリストと、オリジナルのMRリストと、クリーンアップ有意係数リストと、クリーンアップ非有意近傍リストとが入力される。リスト・メモリ内のオリジナルのSPリストは、前回の符号化ステップ314中更新された有意タグを用いてソートされ（1204）、2つのリストに変えられる。オリジナルのSP有意係数リストは、オリジナルのSPリストの位置であって、現在のビットプレーン内の対応する有意係数を持つものとしてタグが付けられた位置を有する。後者の位置は新しいオリジナルMRリストに追加される。オリジナルのSP非有意係数リストは、オリジナルのSPリストの位置を有する。該位置のタグは、現在のビットプレーン内の対応する非有意係数を持つ位置を示す。この後者の位置は、新しいオリジナルのSPリストに追加される。同様に、新しいSPリストは、前回の符号化ステップ314中に生成された有意タグを用いてソートされ（1206）、2つのリストに変えられる。新しいSP有意係数リストは、新しいSPリストの位置であって、現在のビットプレーン内の対応する有意係数を持つものとしてタグが付けられた位置を有する。後者の位置は、新しいMRリストに追加される。新しいSPリストの位置を有する新しいSP非有意係数リストであって、これらの位置を示すタグを持つオリジナルのSP非有意係数リストは、現在のビットプレーン内の対応する非有意係数を持つ。後者の位置は新しいSPリストに追加される。

【0105】オリジナルのSP非有意係数リストの位置と、オリジナルのSP有意係数リストと、新しいSP非有意係数リストと、新しいSP有意係数リスト

と、オリジナルのMRリストと、クリーンアップ有意係数リストと、クリーンアップ非有意近傍リストとは、すべてのサブリストと一緒に併合する、2つの同一の「比較と併合」処理1208と1210の中を通る。第1の「比較と併合」処理1208は、オリジナルのSP有意係数リストの位置と、新しいSP有意係数リストと、オリジナルのMRリストと、クリーンアップ有意係数リストとを比較し併合する（本明細書では、以後第1サブリストと呼ぶ）。第2の「比較と併合」処理1210は、オリジナルSP非有意係数リストの位置と、新しいSP非有意係数リストと、クリーンアップ非有意近傍リストとを比較し併合する（本明細書では、以後第2サブリストと呼ぶ）。「比較と併合」処理1208と1210とは、クリーンアップ符号化ステップ318のループ1104～1114によって現在処理されているクリーンアップ・リストの現在位置も入力として受け取る。

【0106】「比較と併合」処理1210は、第2のサブリストの中の各1つから1つの位置を受け入れ、それらと比較して、どれがJPEG2000の走査順で最も早い位置に在るかを調べる。次いで、「比較と併合」処理1210は、ステップ1211へその結果を出力し、次いで、選択されたサブリストから次の位置を受け入れる。2つのサブリストが同じ位置を持っている場合、双方とも取り除かれ、1つだけが出力される。「比較と併合」処理1208は、「比較と併合」処理1210と同様に実行されるが、第1のサブリストからその位置を受け入れる。このソーティング処理は、1クロック・サイクル当たりのリスト当たり少なくとも1つの位置を生成することで、したがってスループットは、サイクル当たり1係数/ビットのままであることが望ましい。しかし、ループ1104～1114によって処理される現在の座標が、これら全ての「比較と併合」処理が出力しようとしている座標より前に在る場合、この「比較と併合」処理は機能が停止する。この処理によって、新しいオリジナルSPとMRリスト内の座標の欠落が防止される。

【0107】実際には有意である非有意係数の位置が、第2のサブリストに存在する場合もある。これらの位置は、ステップ1211中、比較と併合処理1210の出力から取り除かれる。このステップ1211中、比較と併合処理1210によって出力される各位置は、有効値テーブル1213内のその位置における対応する係数の現在の有意状態と比較される。当該係数が有意であることが有効値テーブル1213によって示された場合、仮定上非有意係数の位置は、ステップ1211において取り除かれる。一方、当該係数が非有意であることが有効値テーブル1213によって示された場合、比較と併合処理1210によって出力された位置は取り除かれず、新しいオリジナルのSPリストの一部を形成する。

【0108】比較有意ステップ1211と比較と併合処理1208とは、新しいオリジナルのSPリストと、新しいオリジナルのMRリストとをそれぞれ出力し、これらのリストは後続するSP符号化パス314とMR符号化パス316で使用される。新しいオリジナルSPリストとMRリストとを生成するステップ1116は、リストの完了後、1212で終了する。

【0109】本構成の変更例では、新しいオリジナルのSP/MRリスト生成処理は継続し、次のビットプレーンのSP符号化パス314に入る。次のSP符号化パス314が始まるときまでに、新しいオリジナルのSPリストの先頭が準備ができていたことが望ましい。それにより機能停止が不要となる。しかし、クリーンアップ符号化パス処理に起因する機能停止が生じないので、リスト生成処理は最大レートで進むはずである。

【0110】本構成のさらなる変更例では、オリジナルのSPリストと新しいSPリストのタグなしで済ませることが可能となる。現在のビットプレーンで有意であるかどうかを示す有効値テーブルから、これらのリストに格納された位置に対応する係数の有効値を読み出すことにより、リスト生成ステップ1116中、これらのリストをソートすることができる。更なる変更例では、リスト・メモリ内に、ビットが有意か否かを示す1ビットを設けて、その係数が以前に処理されたものであれば、このビットが書き込まれる。有効値テーブルへのアクセスを減らすという理由でこのアプローチは好適である。上記ビットはリスト・メモリとして同じメモリ内に在ってもよいし、或いは、別個のメモリ内に在ってもよい。

【0111】上述のように、クリーンアップ符号化パス318の完了後、本方法は任意のさらなるビットプレーンの処理を行うために判定ブロック310へ戻る。判定ブロックが真（イエス）を返した場合、即ち、処理の対象となる更なるビットプレーンが存在しない場合、本方法は終了する（300）。

【0112】次に図15に目を転じると、図3に図示のエントロピ符号化法を実行するためのエントロピ・コーデックが示されている。このエントロピ・コーデックは、リスト・メモリ・マネージャ1502と、ビットプレーン・スプリッタ1504と、ビットプレーン・メモリ1508と、有効値テーブル1510と、リスト・メモリ1506と、コンテキスト生成論理回路1512と、算術符号化部（図示せず）とを有する。リスト・メモリ・マネージャ1502は、専用集積回路の形をとり、その機能は図3のエントロピ符号化法の演算と関連づけられるリスト作成を実現することである。リスト・メモリ・マネージャ1502によって作成されたこれらのリスト（例えば新しいSPリスト）は、リスト・メモリ・マネージャ1502によって、半導体メモリの形のリスト・メモリ1506内に格納される。リスト・メモリ・マネージャ1502は、図3のエントロピ符号化法

自体のコンテキスト生成ステップと符号化ステップとを実行せず、コンテキスト生成論理回路1512とコーデック（図示せず）によって行われる、これらの操作の制御を行う。同様に、リスト・メモリ・マネージャ1502は、ビットプレーン分割操作304と操作306とを行わず、ビットプレーン・スプリッタ1504によって実行される、これらの操作の制御を行う。

【0113】コーデック1500は以下のように図3のエントロピ符号化法を実行する。コード・ブロックのウェーブレット係数がDWTユニット（図示せず）から入力されている間、このコード・ブロックを構成するビットプレーンは、ビットプレーン・スプリッタ1504によってそれぞれのビットプレーン・メモリ1508の中へ書き込まれる。同時に、ビットプレーン・スプリッタ1504は、各係数が有意となり始めるビットプレーンを決定し、有効値テーブル1510内にその情報を格納する。また同時に、ビットプレーン・スプリッタ1504は、どれがビットプレーン内で有効ビットを持つ最上位ビットプレーンであるかを決定し、リスト・メモリ・マネージャ1502へ最上位ビットプレーン番号を送出する。リスト・メモリ・マネージャ1502はまた、符号化の対象となる最下位ビットプレーンのビットプレーン番号もフェッチする。リスト・メモリ・マネージャ1502は、最上位ビットプレーンから始まり、次いで、各ビットプレーンについて、シグニフィカンス・プロパゲーション・パス314のオペレーションと、マグニチュード・リファインメント・パス316と、クリーンアップ・パス308または318とを符号化の対象となる最下位ビットプレーンまで実行する。各パスで、リスト・メモリ・マネージャ1502は、そのパスで符号化／復号を行う必要がある全ての係数である、リスト・メモリ1306の中に格納されている全ての係数の位置リストを作成する。シグニフィカンス・プロパゲーション・パス314の場合、リスト・メモリ・マネージャ1502がオリジナルのSPリストの中を通るとき、新しいSPリストが生成される。

【0114】リスト・メモリ・マネージャ1502は、符号化の対象となるシンボルの位置を決定するとすぐに、有効値テーブル1510から読み出してコンテキスト生成論理回路1512上へ渡すべきその位置の3×3の近傍内の係数の有意値を指示する。リスト・メモリ・マネージャ1502はまた、現在のビットプレーン内のその位置におけるシンボルをビットプレーン・メモリ1508から検索する。次いで、コンテキスト生成論理回路1512は、この有効値情報に基づいてコンテキストを計算し、算術符号化部（図示せず）へその情報を渡す。また、現在のビットプレーン内のその位置におけるシンボルは算術符号化部（図示せず）上へ渡される。

【0115】次に図16に目を転じると、コード・ブロックの変換係数のシンボルのエントロピ符号化を行う別

の方法1600を示すフローチャートが示されている。シンボルのエントロピ符号化を行うこの方法1600は、図3を参照しながら説明したシンボルの符号化を行う方法より単純である。しかし、この方法1600は係数のうちの数ビットを予め知っている必要があるので、エントロピ復号化を行うために、この方法を用いることはできない。したがって、この方法は、単純ではあるものの、エントロピ符号化時にしか利用することができない。

【0116】本方法1600はステップ1602から始まり、このステップで必要なパラメータが全て初期化される。初期化段階1602中、本方法1600は有効値テーブルを生成する。この有効値テーブルは、符号化の対象となるコード・ブロックの係数の有意状態の2次元配列を有する。本方法1600は、コード・ブロック全体を処理し、各係数が有意となり始める位置に在るビットプレーンを決定し、有効値テーブル内にその情報を格納する。次いで、本方法1600は、その係数と関連する有効値テーブルに格納されるビットプレーン番号に関して、現在のビットプレーンにおける係数の有意状態を決定することができる。例えば、有効値テーブル内に、ある係数と関連づけられて格納されている現在のビットプレーンがビットプレーン番号より大きい場合、その係数はその現在のビットプレーンについて非有意となる。現在のビットプレーンがビットプレーン番号より大きくない場合にはその係数は有意になる。このビットプレーン番号は、コード・ブロックの範囲内の係数位置に対応する配列を示す(行、列)座標と共に、配列として有効値テーブルの中に格納される。

【0117】また初期化段階1602中、本方法は、シグニフィカンス・プロパゲーションリスト(SPリスト)と、マグニチュード・リファインメント・リスト(MRリスト)と、クリーンアップ・リスト(Nリスト)として本明細書で名付けられた3つのリストを生成する。これらのリストには、現在のパス中、符号化されるコード・ブロックの範囲内のビット・シンボルの位置が含まれる。例えば、このシグニフィカンス・プロパゲーションリストには、現在のシグニフィカンス・プロパゲーション・パスで符号化されるコード・ブロックの範囲内の全てのビット・シンボルの位置リストが含まれる。‘ビット・シンボルの位置’という用語は、そのビット・シンボルが一部を形成する変換係数のコード・ブロックの範囲内の配列位置を、本明細書では意味する。さらに、これらのリストの各々の範囲内の位置は、JPEG2000の走査順に従って索引化される(図2参照)。最初、シグニフィカンス・プロパゲーションリスト(SPリスト)とマグニチュード・リファインメント・リスト(MRリスト)とは空である。一方、クリーンアップ・リスト(Nリスト)は、コード・ブロックの範囲内の全ての位置をリストし、ノンゼロ・ビットを持つ

最上位ビットプレーンの符号化を行うために使用される。

【0118】開始1602後、コード・ブロックの変換係数が受け取られ、ビットプレーンに分割される(1604)。次のステップ1606では、コード・ブロックのビットプレーンがチェックされ、どのビットプレーンが最上位ノンゼロ・ビットを所有するかが判定される。

【0119】次いで、本方法はステップ1608へ進み、そこで最上位ビットを持つビットプレーンは、1回の単一クリーンアップ・パスでエントロピ符号化される。具体的には、最上位ノンゼロ・ビットを持つビットプレーンに属するビット・シンボルが各々エントロピ符号化される。これらのビット・シンボルは、例えば、図2に図示のように、JPEG2000の走査順でエントロピ符号化される。エントロピ符号化ステップ1608は、符号化対象ビット・シンボルと、そのビット・シンボルのコンテキストとを入力として受け取り、該コンテキストはJPEG2000に従って好適に決定される。

【0120】ステップ1608の完了後、本方法は判定ブロック1610へ進み、そこで処理の対象となるビットプレーンがさらに存在するかどうかのチェックが行われる。もし存在すれば、本方法はステップ1612へ進み、そこで次のビットプレーンのビット・シンボルが検索される。

【0121】ステップ1612の完了後、本方法1600はステップ1613へ進む。ステップ1613中、本方法1600は、まずシグニフィカンス・プロパゲーションリスト(SPリスト)と、マグニチュード・リファインメント・リスト(MRリスト)と、クリーンアップ・リスト(Nリスト)とをクリアする。その後、ステップ1613では、コード・ブロックの範囲内のすべての位置をソートし、シグニフィカンス・プロパゲーションリスト(SPリスト)と、マグニチュード・リファインメント・リスト(MRリスト)と、クリーンアップ・リスト(Nリスト)との中にこれらの位置を格納する。ソート・ステップ1613は、これらのリストのうちの1つだけに各位置が格納されるように、これらの位置を格納する。これらの格納された位置は、コード・ブロック内の対応する係数の(行、列)座標を示す。

【0122】ソート・ステップ1613中、本方法はJPEG2000の走査順にコード・ブロックの全ての位置に互って増分を行う。

【0123】現在の位置走査中、ソート・ステップ1613は、現在走査中の位置を取り囲む3×3の近傍の位置に対応する、有効値テーブルに格納されたビットプレーン番号を検索する。ソート・ステップはまた、現在走査中の位置に対応する有効値テーブル内に格納されたビットプレーン番号も検索する。上述のように、有効値テーブルに格納されたビットプレーン番号は、その係数が有意になる位置に存在するビットプレーンを示す。次い

でソート・ステップ1613は、現在位置の係数の有意状態を含む、 3×3 の近傍の各係数の有意状態を、現在のビットプレーンにおいて決定する。次いで、ソート・ステップ1613は、以下の規則に従って、SP、MRまたはNリストのうちのいずれに、その位置を格納すべきかを判定する。

【0124】規則1：走査中の現在位置 $P(x, y)$ の係数が有意状態 $\geq N$ （但し、 N は処理中の現在のビットプレーン）を持っていれば、現在位置はMRリストに進む。

【0125】規則2：走査中の現在位置 $P(x, y)$ の係数が、 N より小さい（ $< N$ ）有意状態を持ち、かつ、JP EG 2000の走査順で現在位置 $P(x, y)$ の前に在る係数である、現在位置 $P(x, y)$ の 3×3 の近傍内の任意の係数が、有意状態 $\geq N$ （但し、 N は処理中の現在のビットプレーン）を持つ係数を持っていれば、現在位置 $P(x, y)$ はSPリストに入る。

【0126】規則3：走査中の現在位置 $P(x, y)$ の係数が、 N より小さい有意状態（ $< N$ ）を持ち、かつ、JP EG 2000の走査順で現在位置 $P(x, y)$ の後に在る現在位置 $P(x, y)$ の 3×3 の近傍内の任意の位置であって、有意状態 $> N$ （但し、 N は処理中の現在のビットプレーン）を持つ位置を持っていれば、現在位置 $P(x, y)$ はSPリストに入る。

【0127】規則4：走査中の現在位置 $P(x, y)$ がMRリスト又はSPリストに格納されていなければ、その位置はNリストに格納される。即ち、クリーンアップ（N）リストには、SPリストとMRリストにない位置が含まれる。

【0128】このようにして、ソート・ステップ1613は、係数の位置をソートしてビットプレーンN用のSP、MR、Nリストの中へ入れる。次いで、これらのリストを送って、シグニフィカンス・プロパゲーション・パスと、マグニチュード・リファインメント・パスと、クリーンアップ・パスとの中で符号化を行うことができる。

【0129】ソート・ステップ1613の完了後、本方法はステップ1614へ進み、現在のビットプレーンのシグニフィカンス・プロパゲーション・パス内で関連するビット・シンボルの処理が行われる。このステップ1614で、ステップ1613で生成されたビット・シンボルである、現在のSPリスト内の位置に対応する現在のビットプレーン内のビット・シンボルがエントロピ符号化される。

【0130】SP符号化パス1614の完了後、本方法はマグニチュード・リファインメント符号化パス1616へ進む。このステップ1616で、ステップ1613で生成されたビット・シンボルである、現在のマグニチュード・リファインメント・リスト内の位置に対応する現在のビットプレーン内の全てのビット・シンボルがエントロピ符号化される。

【0131】MR符号化パス1616の完了後、本方法はクリーンアップ符号化パス1618へ進む。クリーンアップ符号化ステップ1618で、（前回の符号化パス中生成されたビット・シンボルである）、現在のクリーンアップNリスト内の位置に対応する、現在のビットプレーン内の全てのビット・シンボルがエントロピ符号化される。

【0132】クリーンアップ符号化パス1618の完了後、本方法は判定ブロック1610へ戻り、任意の更なるビットプレーンの処理が行われる。判定ブロックが真（イエス）を返した場合、即ち、処理の対象となる更なるビットプレーンが存在しない場合、本方法は終了し（1620）、コール方法へ戻る。判定ブロックが真（イエス）を返さなかった場合、本方法は、次のビットプレーンの処理を行うためにステップ1612へ進む。好適には、本方法は所定の最下位ビットプレーンまでビットプレーン処理を続けることが望ましい。

【0133】好適には、本方法が全ての3つのリスト、即ち、SP、MR及びNリストを生成することが望ましい。本構成の変更例では、本方法はSP、MR及びNリストのうちの少なくとも1つのリストを生成することができる。この場合、本方法はリストを持っていない任意のパス中コード・ブロック全体を走査する。

【0134】次に図17に目を転じると、図16に図示のエントロピ符号化法を実行するエントロピ符号器が示されている。エントロピ符号器1700は、図16を参照しながら上述した規則に従って現在のビットプレーンの係数の位置をソートする係数ソータと、操作1604と1606とを行うためのビットプレーン・スプリッタ（図示せず）と、コード・ブロックの各ビットプレーンのビット・シンボルを格納するためのビットプレーン・メモリ（図示せず）と、有意値を格納するための有効値テーブル1710とを有する。エントロピ符号器はまた、係数ソータ1702によって生成されるSPリストを格納するためのSPリスト・メモリ1704と、係数ソータ1702によって生成されるMRリストを格納するためのMRリスト・メモリ1706と、係数ソータ1702によって生成されるクリーンアップ・リストを格納するためのクリーンアップ・リスト・メモリ1708をも有する。エントロピ符号器は、図16を参照しながら説明したように、コンテキスト生成論理回路1712と、SPリストと、MRリストと、クリーンアップ・リスト内に、それらのリストのそれぞれのパス中格納された位置に対応するビット・シンボルを符号化するための算術符号化部（図示せず）とを更に有する。SPリスト・メモリ1704は、ダブルバッファ構成である。係数ソータ1702が、ビットプレーン n のSPリスト用の位置を生成して、それらの位置をSPリスト・メモリ1704内に格納している間、コンテキスト生成論理回路1712はビットプレーン $(n+1)$ 用の完成したSP

リストを読み出している。MRリスト・メモリ1706とクリーンアップ・リスト・メモリ1708とはまた、SPリスト・メモリ1704と同様の方法でダブルバッファとしても機能する。

【0135】図3と図16を参照しながら説明した前述の好ましい方法は、ある特別の制御フローを有する。これらの方法には、本発明の精神または範囲から逸脱することなく様々な制御フローを用いる他の多くの変形例が存在する。例えば、図3のエントロピ符号化法は、いくつかの並列操作を部分的に有する。図3のエントロピ符号化法を改変して、当業者には明らかであるようなシーケンシャルな操作であるかのように、これらの並列操作を実現することもできる。後者（シーケンシャルなオペレーション）の場合、図3のこの改変されたエントロピ符号化法は、汎用コンピュータのソフトウェアとしての実現に適している。例えば、前処理ステップ412と後処理ステップ418とは、いくつかの改変例についてはシーケンシャルに機能してもよい。汎用コンピュータのソフトウェアとして図16のエントロピ符号化法を実現してもよい。

【0136】次に図18に目を転じると、前述の方法を実行することができる汎用コンピュータが示されている。コンピュータ・システム1800の範囲内で実行するアプリケーション・プログラムのようなソフトウェアとして、これらの方法の処理を実行してもよい。特に、前述の方法のステップは、コンピュータによって実行されるソフトウェア内の命令によって実行される。例えば、以下説明する記憶装置を含むコンピュータ可読媒体にこのソフトウェアを格納してもよい。このソフトウェアは、コンピュータ可読媒体からコンピュータの中へロードされ、次いで、コンピュータによって実行される。コンピュータ可読媒体に保存されるようなソフトウェアまたはコンピュータ・プログラムを持つコンピュータ可読媒体は、コンピュータ・プログラム製品である。

【0137】コンピュータ・システム1800は、コンピュータ・モジュール1801と、キーボード1802とマウス1803等の入力装置と、プリンタ1815と表示装置1814とを含む出力装置とを有する。コンピュータ・モジュール1801は、電話回線1821またはその他の機能媒体を介して接続可能な通信ネットワーク1820と交信するための変復調（モデム）送受装置1816を使用する。モデム1816を使用して、インターネットやローカル・エリア・ネットワーク（LAN）や広域ネットワーク（WAN）等の他のネットワーク・システムにアクセスすることもできる。

【0138】コンピュータ・モジュール1801には、少なくとも1つのプロセッサ・ユニット1805と、例えば半導体ランダム・アクセス・メモリ（RAM）やリード・オンリ・メモリ（ROM）から形成される記憶装置（メモリ）1806と、ビデオ・インターフェース1

807を含む入力/出力（I/O）インターフェースと、キーボード1802とマウス1803およびオプションとしてジョイスティック（例示せず）用のI/Oインターフェース1813と、モデム1816用のインターフェース1808とが一般に含まれる。記憶装置1809が提供され、この装置には一般にハードディスク・ドライブ1810とフロッピー（登録商標）ディスク・ドライブ1811とが含まれる。磁気テープ駆動装置（例示せず）を使用することもできる。不揮発性データソースとしてCD-ROM駆動装置1812が一般に設けられる。コンピュータ・モジュール1801の構成要素1805～1813は、一般に、相互接続バス1804を介して、また、コンピュータ・システム1800の従来の動作モードで結果として得られる当業者に公知の方法で交信を行う。実施の形態の実行が可能なコンピュータの例には、IBM-PCおよび互換機、Sun Sparstationやそこから進化した同様のコンピュータ・システムが含まれる。

【0139】一般に、前述の方法のアプリケーション・プログラムはハードディスク・ドライブ1810に常駐し、プログラムの実行時にプロセッサ1805によって読み出され、制御される。半導体メモリ1806を用いて、おそらくハードディスク・ドライブ1810に従って、ネットワーク1820からフェッチされたプログラムと任意のデータの中間の格納を行うことができる。いくつかの例では、CD-ROMやフロッピーディスクに符号化され、対応する駆動装置1812や1811を介して読み出されるアプリケーション・プログラムや、或いは、ユーザーがネットワーク1820からモデム装置1816を介して読み出すことができるアプリケーション・プログラムをユーザーへ供給することができる。さらに、磁気テープ、ROMあるいは集積回路、光磁気ディスク、コンピュータ・モジュール1801と別の装置との間の無線伝送チャネルまたは赤外線伝送チャネル、PCMCIAカードのようなコンピュータ可読カード、ウェブ等に保存されるeメール伝送と情報を含むインターネットとイントラネットを含む他のコンピュータ可読媒体からコンピュータ・システム1800の中へソフトウェアをロードすることもできる。上述のものは単に関連するコンピュータ可読媒体の例示にすぎない。本発明の範囲と精神から逸脱することなく、他のコンピュータ可読媒体を実際に使用することも可能である。

【0140】本発明の実施の形態がコンピュータ産業とビデオ画像産業、並びに、ビデオ及びカメラ産業のような関連分野に適用可能であることは、以上のことから明らかである。

【0141】上述のことは、本発明のいくつかの実施の形態について説明するものにすぎず、本発明の範囲と精神から逸脱することなく改変及び/又は変更を行うことが可能であり、以上の実施の形態は例示的なものであ

て、限定なものではない。

【図面の簡単な説明】

【図1】従来技術のJ P E G 2 0 0 0で利用される係数“X[m, n]”の8個の包囲係数の近傍有意状態を示す図である。

【図2】従来技術のJ P E G 2 0 0 0で利用されるコード・ブロック用のコード・ブロック走査パターンの一例を例示する概略ブロック図である。

【図3】第1の構成に従うエントロピ符号化法を示すフローチャートである。

【図4】図3に図示のエントロピ符号化法で用いられる最初のクリーンアップ符号化ステップ308の詳細処理を示すフローチャートである。

【図5】図4に描かれるようなi番目のストリップのj番目の列の前処理ステップ412をさらに詳細に示すフローチャートである。

【図6】図4に描かれるような(i-1)番目のストリップの(j-2)番目の列の後処理ステップ418をさらに詳細に示すフローチャートである。

【図7A】、

【図7B】図3のシグニフィカンス・プロバゲーション(SP)符号化ステップ374をさらに詳細に示すフローチャートである。

【図8】SP符号化ステップ314のステップ720をさらに詳細に示すフローチャートである。

【図9】SP符号化ステップ314のクリーンアップ・

リスト生成ステップ708をさらに詳細に示すフローチャートである。

【図10】マグニチュード・リファインメント符号化パス316をさらに詳細に示すフローチャートである。

【図11】図3のクリーンアップ符号化パス318をさらに詳細に示すフローチャートである。

【図12】新しいオリジナルのSPリストと新しいMRリストとを生成するためのステップ1116をさらに詳細に示すフローチャートである。

【図13A】、

【図13B】図7Aのステップ732によって渡される新しいSPリストに非有意近傍を加える処理を示すフローチャートである。

【図14】図13Aと図13Bの前処理プロセスとともに使用する後処理プロセスを示すフローチャートである。

【図15】図3に図示のようなエントロピ符号化法を実現するためのエントロピ・コーデックの概略図を示す図である。

【図16】第2の構成に従うエントロピ符号化法を示すフローチャートである。

【図17】図16に示すエントロピ符号化法を実現するためのエントロピ符号器の概略図である。

【図18】図3と図16の方法を実行する汎用コンピュータの概略図である。

【図1】

Si[m-1,n-1]	Si[m-1,n]	Si[m-1,n+1]
Si[m,n-1]	"X[m,n]"	Si[m,n+1]
Si[m+1,n-1]	Si[m+1,n]	Si[m+1,n+1]

Fig. 1 Prior Art

【図2】

1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31
4	8	12	16	20	24	28	32
33	37	41	45	49	53	57	61
34	38	42	46	50	54	58	62
35	39	43	47	51	55	59	63
36	40	44	48	52	56	60	64

Fig. 2 Prior Art

【図 3】

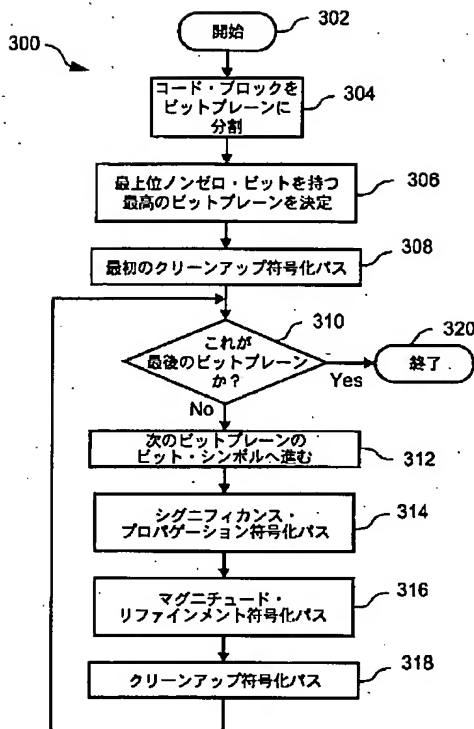


Fig. 3

【図 4】

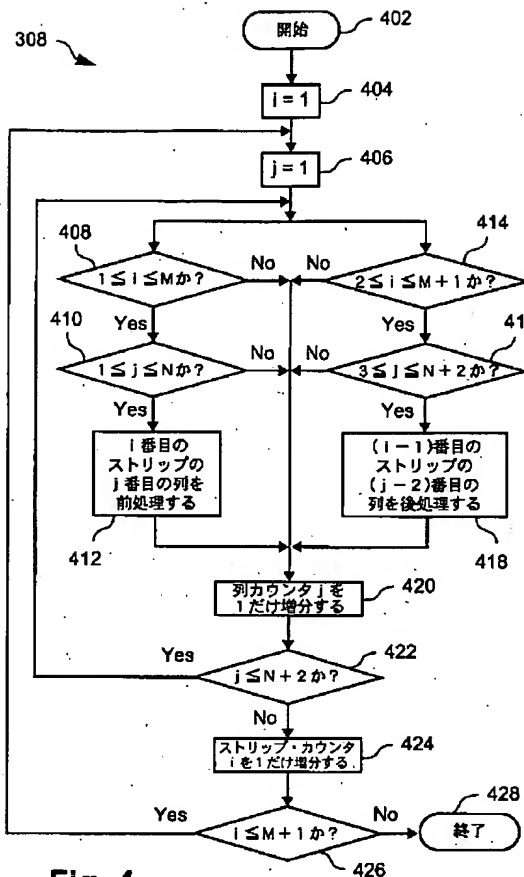


Fig. 4

【図 8】

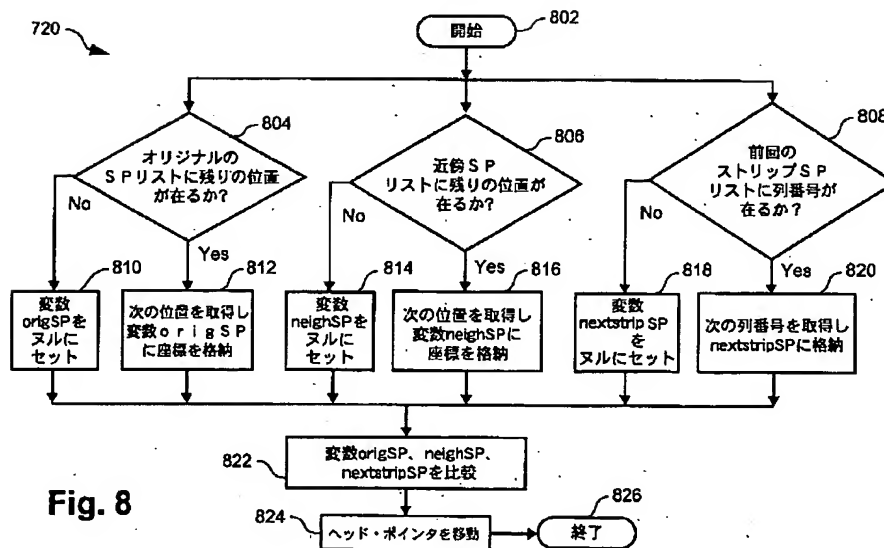
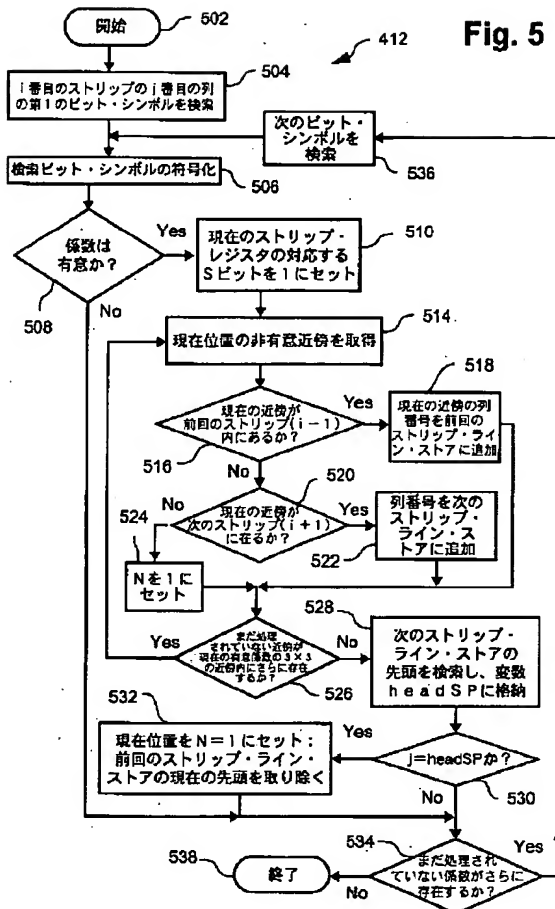
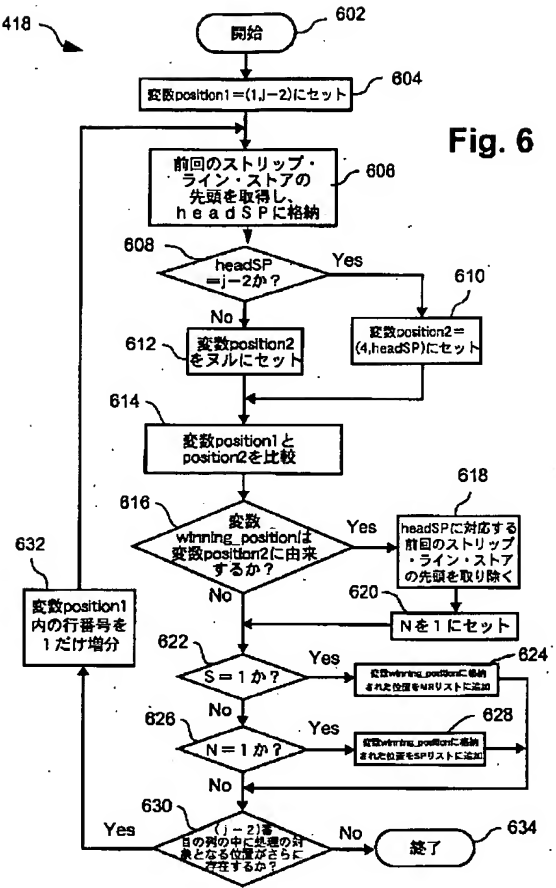


Fig. 8

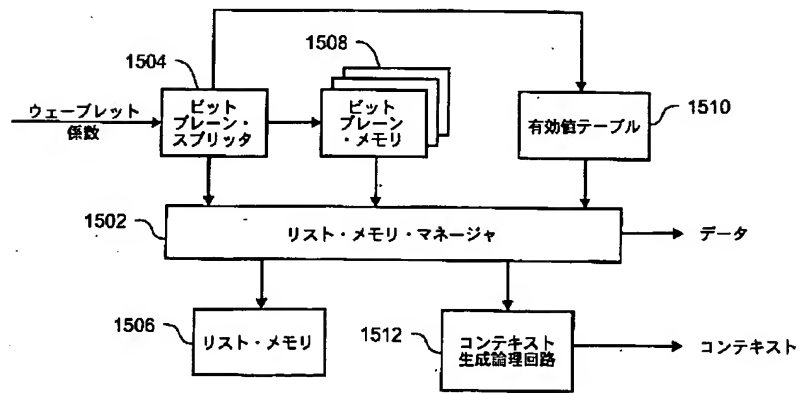
【図5】



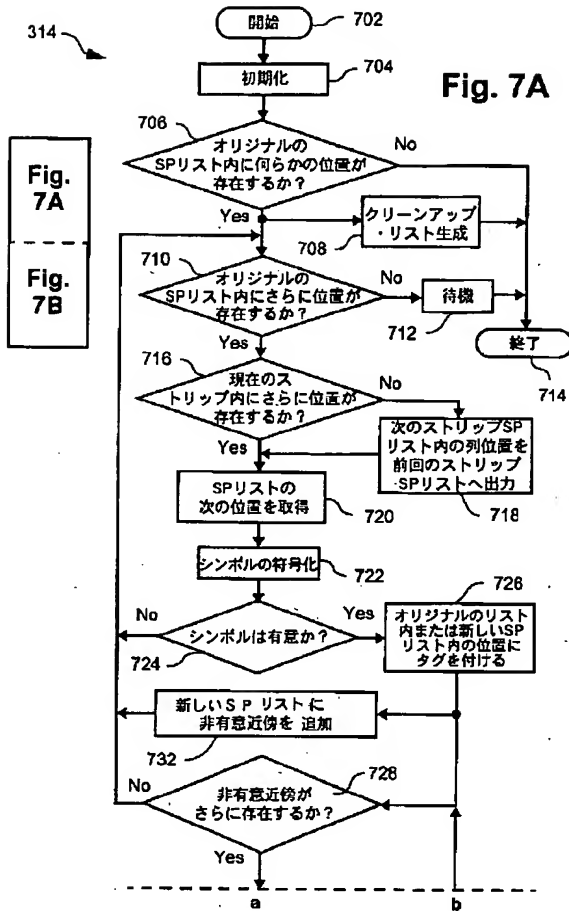
【図6】



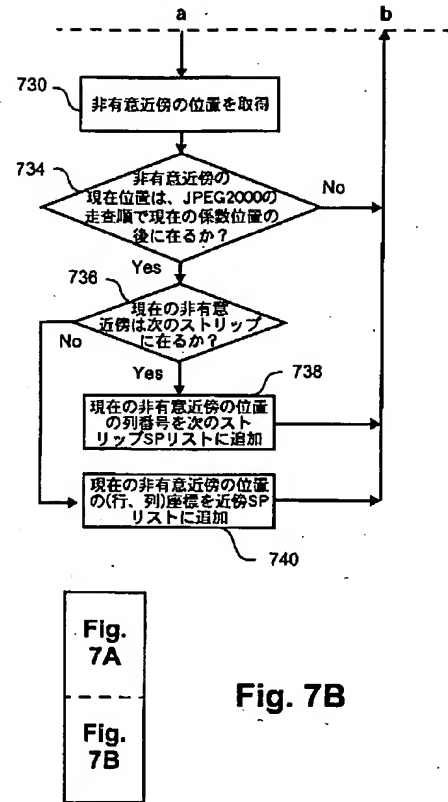
【図15】



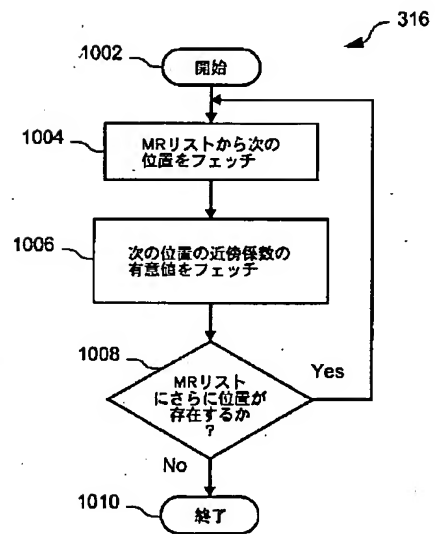
【図 7 A】



【図 7 B】



【図 10】



【図 9】

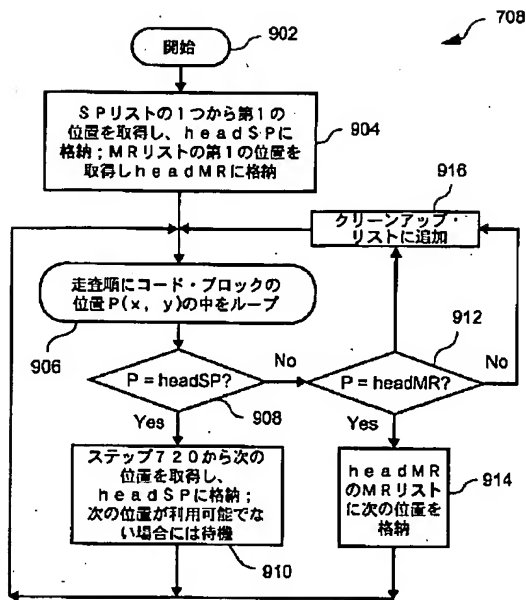


Fig. 9

【図 11】

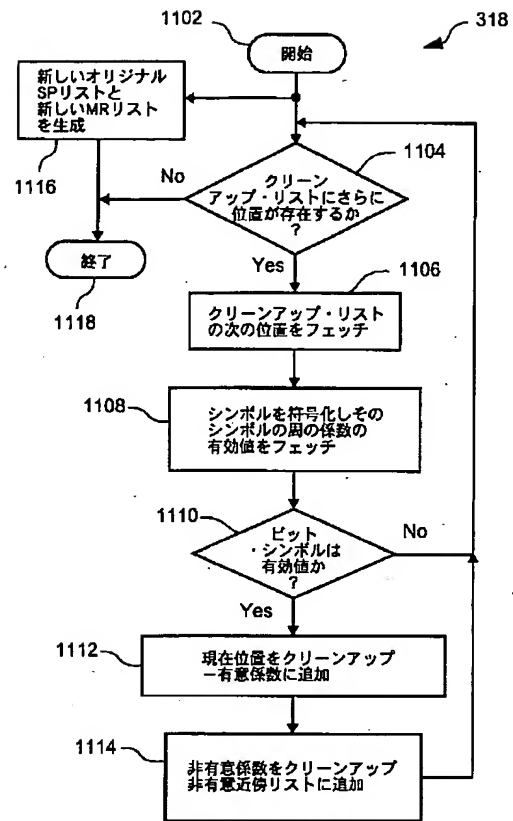


Fig. 11

【図 12】

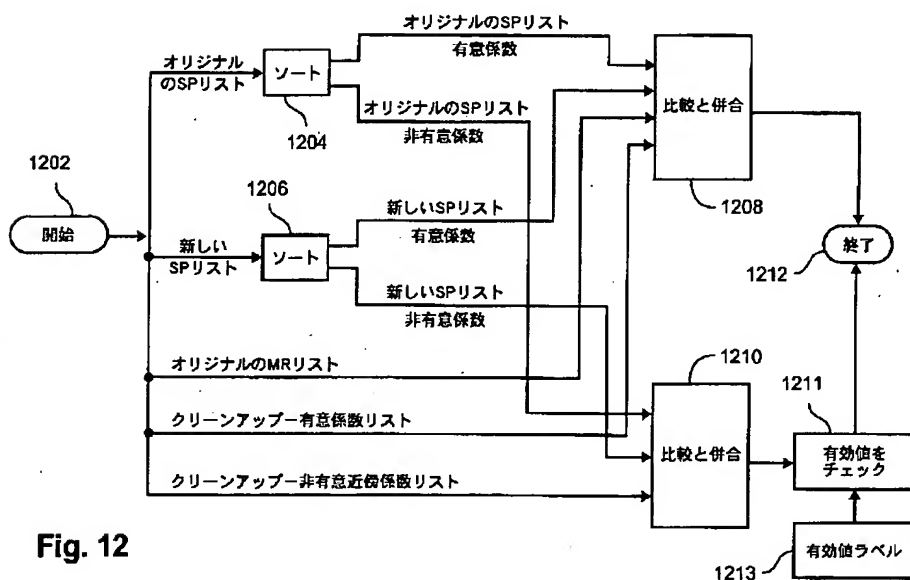
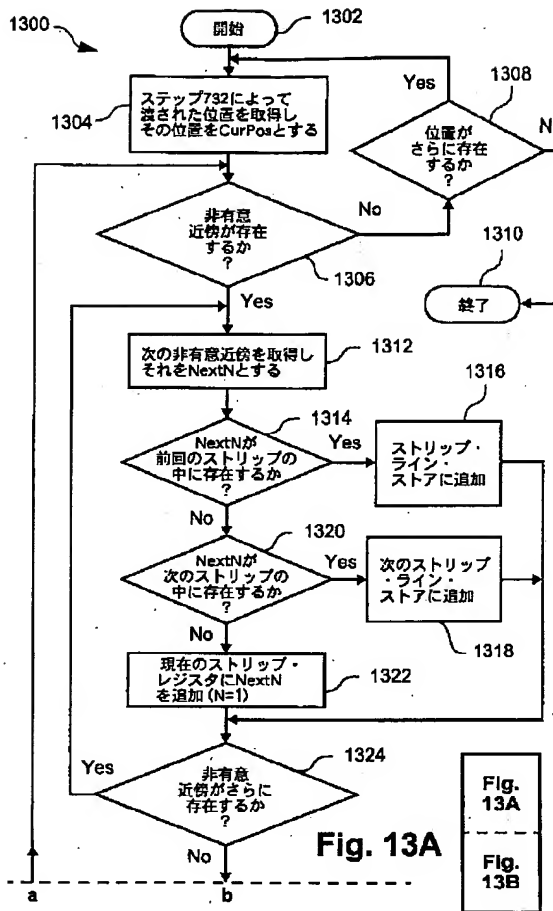
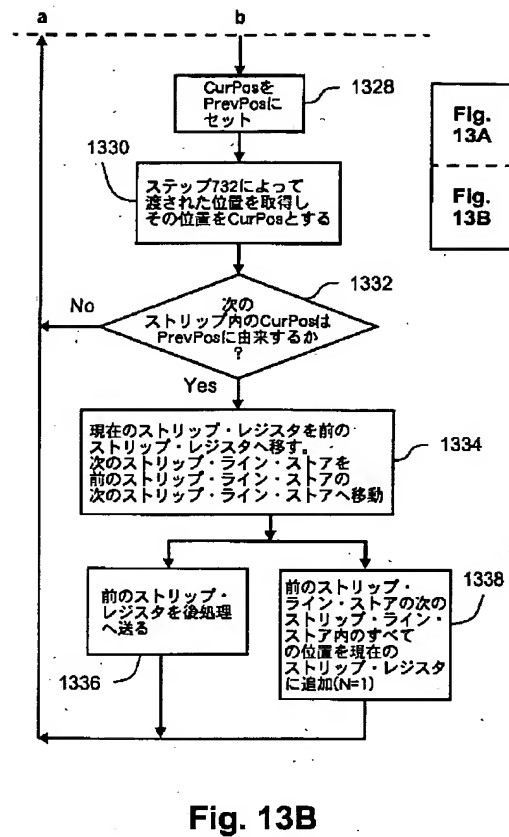


Fig. 12

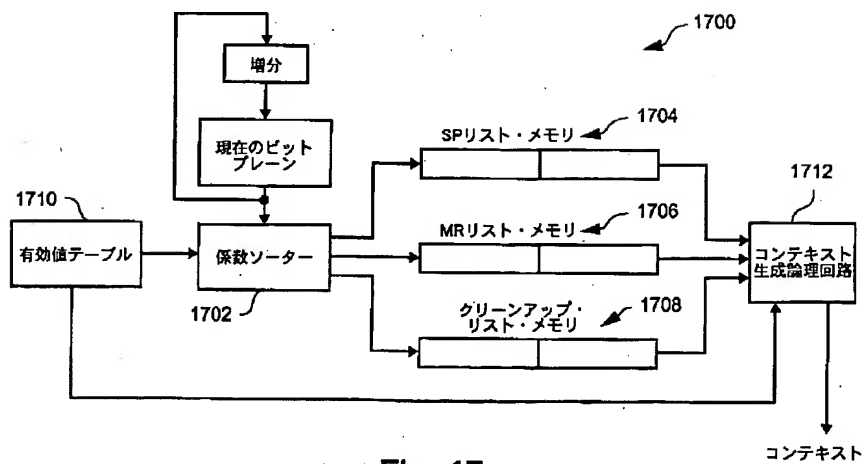
【図13A】



【図13B】



【図17】



【図14】

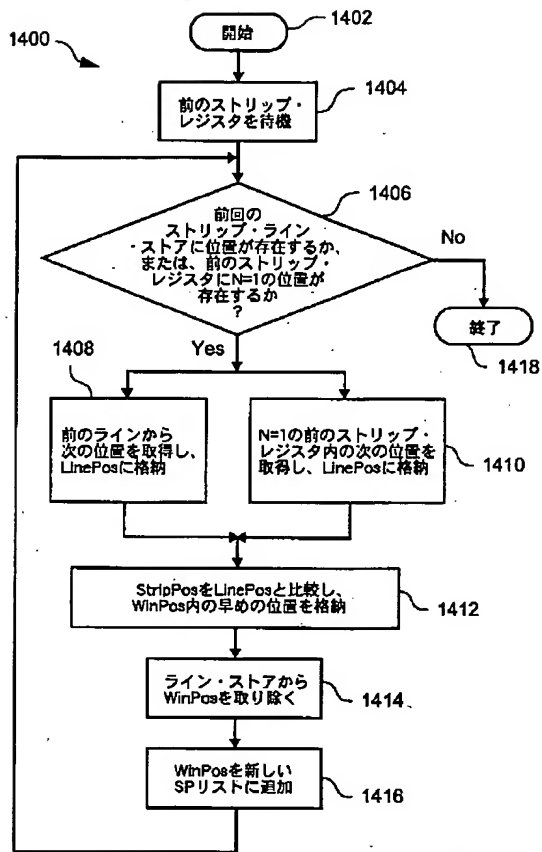


Fig. 14

【図16】

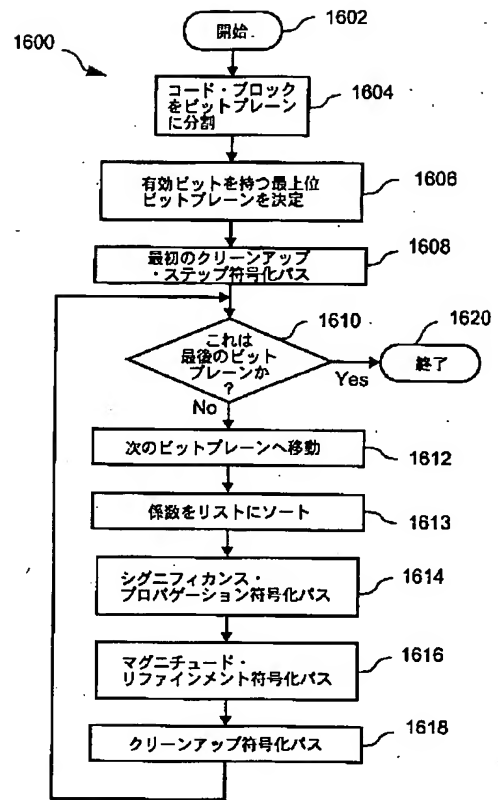


Fig. 16

【図18】

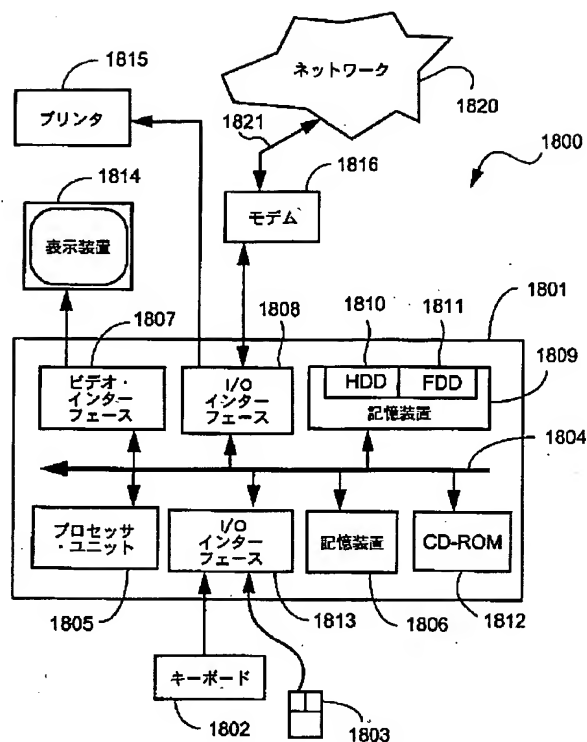


Fig. 18

フロントページの続き

Fターム(参考) 5C059 MA00 MA24 MA35 MD07 ME01
 UA02 UA05 UA15 UA34 UA39
 5J064 BA09 BA16 BC01 BC02 BC04
 BD04